

**Ingeniería de software 2**

**Clave I5900**

**NRC** 153358

**2019 A**

**Sección D04**

**Alumno:**

Índice

Introducción……………………………………………………………………………………………………………….4

Definición del problema………………………………………………………………………………..4

Solución propuesta………………………………………………………………………………………4

Alcances del sistema……………………………………………………………………………………5

Objetivos del sistema………………………………………………………………………………….5

Requerimientos de almacenamiento…………………………………………………………5

Análisis de factibilidad………………………………………………………………………………..5

Visión……………………………………………………………………………………………………………………….7

introduccion……………………………………………………………………………………………….7

positioning…………………………………………………………………………………………………7

stake holder and user presciptions……………………………………………………………8

product overview……………………………………………………………………………………..11

product features……………………………………………………………………………………….12

use specifications…………………………………………………………………………………………………..13

software arquitecture……………………………………………………………………………………………18

introduccion………………………………………………………………………………………………18

arquitectural representation……………………………………………………………………18

arquitectural goals……………………………………………………………………………………19

use case view……………………………………………………………………………………………19

logical view………………………………………………………………………………………………19

deployment view……………………………………………………………………………………22

perfomance……………………………………………………………………………………………23

fase de elaboración……………………………………………………………………………………………..23

diagrama de clases…………………………………………………………………………………23

diagrama de actividad……………………………………………………………………………25

diagrama de secuencia……………………………………………………………………………28

diagrama colaboración…………………………………………………………………………..30

diagrama de componentes…………………………………………………………………….31

Diagrama de despliegue…………………………………………………………………………32

Diagrama ER……………………………………………………………………………………………33

Test plan……………………………………………………………………………………………………………..34

introduccion……………………………………………………………………………………………34

evaluation mission………………………………………………………………………………….35

target test items…………………………………………………………………………………….36

test approach……………………………………………………………………………………………36

entry and exit criteria………………………………………………………………………………47

iteration mielstones…………………………………………………………………………………47

Software development plan…………………………………………………………………………………48

overview…………………………………………………………………………………………………48

Project organization…………………………………………………………………………………48

Management process……………………………………………………………………………..48

calendar………………………………………………………………………………………………….50

Project control…………………………………………………………………………………………51

User manual………………………………………………………………………………………………………..52

Introducción

# Definición del problema

El plantel de CUCEI (Centro Universitario de Ciencias Exactas e Ingenierías) de la Universidad de Guadalajara requiere de un nuevo sistema para su biblioteca.

La biblioteca tiene clasificado a todo en cinco divisiones:

* LIBROS
* EMPLEADOS
* CLIENTES
* TESIS
* SANCIONES

.

Si la biblioteca no lleva un buen manejo de los prestamos puede ocasionar perdida de libros, tesis, etc.., asi como confusión de los usuarios de la bibliotec.

# Solución Propuesta

Este proyecto busca atacar este problema por medio del desarrollo de un Sistema controlar y almacenar la información sobre los libros, asi como los préstamos que se hicieron y las sanciones por entrega tardía.

Cada libro guardado en la biblioteca cuenta con un título, un autor, su editorial, ISBN, la fecha de publicación y el género del libro o a que rama universitaria pertenece. Aparte se mantiene un registro del número de copias en existencia actualmente en la biblioteca para llevar un control de los préstamos y revisar si un libro se puede prestar o no.

Por otra parte, también mantiene un registro de sus empleados en todas las áreas de la biblioteca, cada empleado cuenta con Nombre, Edad, Puesto que desempeña y una ID de empleado. Un empleado tiene la oportunidad de sacar libros de la biblioteca por lo que se utiliza su ID al momento de hacer el préstamo.

Para cada cliente se almacena su nombre, dirección, número de teléfono y una ID de usuario.

Existen ciertas restricciones en la biblioteca tales como:

* Máximo de libros para préstamo: 5
* Máximo de tesis para préstamo: 3
* Máximo de revistas para préstamo: 2
* Ejemplar 1 solo se presta viernes y se entrega lunes
* Ejemplar 2 en adelante préstamo de 8 días

Y las penalizaciones se controlan de la siguiente manera:

* 1 peso por hora para ejemplar 1
* 5 pesos por día para ejemplar 2 en adelante

# Alcance del sistema

Los componentes principales del sistema serán los siguientes:

* Una BD hecha sobre MySQL u otro DBMS.
* Pagina creada con html y php
* Un sistema de administración de prestamos
* Llevar un registro de los libros disponibles

# Objetivos del sistema

El sistema cumplirá con los siguientes objetivos:

* El preciso rastreo del inventario de libros
* Características para facilitar los prestamos a los usuarios
* Rastreo de los prestamos y sanciones
* Consultas en la base de datos
* Consultas de copias de libro
* Consultas de sanciones
* Registro de usuarios

# Revisión de los requerimientos de almacenamiento

El sistema tendrá las siguientes entidades de información:

* Empleados
* Libros
* Prestamos
* Sanciones de préstamo
* Registro de usuarios

# Reporte de Análisis de Factibilidad

Posibles alternativas son las siguientes:

* Alternativa A: Sistema manual, el registro de prestamos es físico
* Alternativa B: Comprar un sf y ajustarlo a las necesidades de la biblioteca
* Alternativa C: Desarrollar el sistema como un proyecto escolar

|  |  |  |  |
| --- | --- | --- | --- |
| Factores de cualidad | Alt-A | Alt-B | Alt-C |
| Eficiencia | 3 | 9 | 9 |
| Seguridad | 3 | 7 | 5 |
| productividad | 4 | 8 | 7 |
| Adaptabilidad | 3 | 7 | 9 |
| Potencial de crecimiento | 2 | 7 | 9 |
| Compatibilidad | 0 | 5 | 5 |
| Confiabilidad | 4 | 8 | 8 |

## Análisis de factibilidad de la alternativa C.

La factibilidad técnica de este sistema se sumaría de la siguiente manera:

* Adquisición del Sw requerido
* Se cuenta con el conocimiento y experiencia requerido
* Se cuenta con la experiencia necesaria para el manejo del SF
* Se cuenta con los empleados para el mantenimiento correcto del SF

La factibilidad operacional de este sistema se sumaría de la siguiente manera:

* La tecnología requerida es relativamente gratuita ya que el software se creará como proyecto escolar
* El tiempo de desarrollo será alto pero los beneficios ganados serán altos.
* E software se supervisado por un profesor.
* No se requerirá entrenamiento para el usuario, solo un manual para que utilice como guía.

El riesgo asociado de esta alternativa puede ser alto o bajo, dependiendo de cómo sea administrado por el estudiante. El estudiante tendrá control total sobre el proyecto.

Calendario del proyecto

|  |  |  |
| --- | --- | --- |
| Actividad# | Descripción de actividad | Semanas |
| 1 | Initial System requirement | 2 |
| 2 | Requirement specifiction | 2 |
| 3 | Desing specification | 3 |
| 4 | Data base creation | 2 |
| 5 | Software development | 7 |
| 6 | Software testing | 1 |
| 7 | Software documentation | 1 |
| 8 | Software installation | 1 |
| 9 | Total estimated duration | 19 |

Vision

# Introduction

El propósito de este documento es colectar, analizar y definir características del Sistema de librería. Se enfoca en las necesidades del cliente y los usuarios y por qué este sistema es necesario.

Los detalles de cómo el sistema cumple estas necesidades esta detallado en los casos de uso y especificación suplementarias.

## Purpose

Analizar y definir las necesidades y caracteristicas del sistema, asi como dar a entender por que es necesario el sistema

# Positioning

## Business Opportunity

Con este Proyecto se intenta lograr crear un sistema de fácil uso para el cliente , así como dar una buena impresión de nuestra manera de trabajo y atención a los detalles para que el cliente este satisfecho.

## Problem Statement

[Provide a statement summarizing the problem being solved by this project. The following format may be used:]

|  |  |
| --- | --- |
| The problem of | La biblioteca no tiene un Sistema para llevar un manejo de empelados ni el manejo de prestamos |
| affects | Afecta a la escuela y usuarios |
| the impact of which is | Esperamos que con este sistema la administración y manejo se facilite para aquellos que la administran |
| a successful solution would be | Crear un Sistema de facil uso para llevar un buen manejo de los empleados y control de la biblioteca |

## Product Position Statement

[Provide an overall statement summarizing, at the highest level, the unique position the product intends to fill in the marketplace. The following format may be used:]

|  |  |
| --- | --- |
| For | Administradores y usuarios de la biblioteca |
| Who | Necesitan un Sistema de ,manejo de empleados y manejo de la biblioteca |
| The (product name) | Sistema de administración de biblioteca |
| That | Ayudara a la escuela a llevar una mejor administración |
| Unlike | Sistema creado por escuela |
| Our product | [statement of primary differentiation] |

[A product position statement communicates the intent of the application and the importance of the project to all concerned personnel.]

# Stakeholder and User Descriptions

## Market Demographics

• cuál es la reputación de tu organización en el mercado?

La organización no tiene una reputación establecida.

• cuál te gustaría que fuera?

Que la organización sea percibida como una organización de confianza con una buena metodología de trabajo

• como este producto beneficiara tus

Este proyecto ayudara a la organización a establecerse como una compañía de confianza , cumpliendo los requisitos del cliente y dando a notar que el producto es de calidad

## Stakeholder Summary

[There are a number of stakeholders with an interest in the development and not all of them are end users. Present a summary list of these non-user stakeholders. (The users are summarized in section 3.3.)]

|  |  |  |
| --- | --- | --- |
| Name | Description | Responsibilities |
| tecno | Compañia encargada del desarrollo del sistema | * Asegura que el Sistema cumplira las caracteristicas * Llevar un segimiento de la vida del proyecto * Tener una Buena comunicacion con los clientes * aClarar dudas que puedan surgir durante la realizacion del proyecto |

## User Summary

[Present a summary list of all identified users.]

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Responsibilities | Stakeholder |
| Gerente | Princiapal usuario del sistema | * Capturar informacion de los empleados * Modificacion de la informacion de los empleados * Altas y bajas de empleados * Registro de usuarios * Manejo de prestamos * Registro de productos * Actualización de DB |  |

## User Environment

[Detail the working environment of the target user. Here are some suggestions:

* Múltiples personas involucradas, administrador y usarios
* EL Sistema se creará como un sitio web

## Stakeholder Profiles

|  |  |
| --- | --- |
| Representative | Stakeholder del Proyecto: luis |
| Description | Encargado del proyecto |
| Type | Experiencia intermedia creando sistemas y páginas web |
| Responsibilities | Levar un seguimiento adecuado del proyecto  Lograr cumplir los requisitos del cliente |
| Success Criteria | Se considerara exitoso si el Proyecto cumple con la mayoria de los requisitos |
| Involvement | Supervisor del proyecto |
| Deliverables |  |
| Comments / Issues |  |

## User Profiles

### 

|  |  |
| --- | --- |
| Representative |  |
| Description | Administrador de la biblioteca |
| Type | Tiene poca experiencia manejando este tipo de sistemas, se requerirá capacitarlo |
| Responsibilities | Lkvar un control de los empleados usando el Sistema creado |
| Success Criteria | Se considerara exitosos si el Sistema cumple con los requisitos que pidio el cliente |
| Involvement | Principal usuario del sitema |
| Deliverables |  |
| Comments / Issues | Mal uso del sistema |

Quedando por lo tanto definidos las características de los usuarios:

|  |  |
| --- | --- |
| Tipo de usuario | Administrador |
| Formación | Cualquier nivel |
| Habilidades | Conocimiento basicos de computacion |
| Actividades | \*Crear nuevos empleados, modificar información, ver información y eliminar.Prestamos, manejo de los productos |

## 

## Key Stakeholder or User Needs

Los usuarios no están capacitados para usar este tipo de sistema:

• What are the reasons for this problem?

No habían utilizado un sistema para manejo de empleados o similar

• How is it solved now?

Se capacitará a los principales usuarios

• What solutions does the stakeholder or user want?]

Una Buena explicación y capacitacion sobre el sistema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Need | Priority | Concerns | Current Solution | Proposed Solutions | |
| Mejor entendimiento del sistema | alta | Mal uso del sistema |  | | Capacitar a los usuarios |

## Alternatives and Competition

El cliente podría ir con otra compañía para la creación del Sistema , también podrían considerar como alternativa llevar registros físicos de la información de sus empleados.

# Product Overview

## Product Perspective

Se plantea un sistema en el cual el administrador pueda llevar un control básico de sus empleados, con fácil acceso a la información básica de estos. Se le dara permisos al administrador de poder realizar funciones básicas con el sistema de empleados, ya sea eliminar, modificar o agregar nueva información.

El administrador también llevara a cabo el registro de usuario de la biblioteca asi como los presamos que estos realicen

## Cost and Pricing

El costo del sistema se ira cambiando mientras se realiza , pero no esperan costos estrax ya que solo es necesario que el cliente tenga acceso a una computadora y conexión a internet

# Product Features

* Cada entidad de información es identificada por un ID de referencia.
* Cada libro guardado en la biblioteca cuenta con un título, un autor, su editorial, ISBN, la fecha de publicación y el género del libro o a que rama universitaria pertenece.
* Aparte se mantiene un registro del número de copias en existencia actualmente en la biblioteca para llevar un control de los préstamos y revisar si un libro se puede prestar o no.
* Por otra parte, también mantiene un registro de sus empleados en todas las áreas de la biblioteca, cada empleado cuenta con Nombre, Edad, Puesto que desempeña y una ID de empleado. Un empleado tiene la oportunidad de sacar libros de la biblioteca por lo que se utiliza su ID al momento de hacer el préstamo.
* Para cada cliente se almacena su nombre, dirección, número de teléfono y una ID de usuario.
* Cada usuario deberá registrarse en la plataforma para poder hacer uso de ella.
* Cada usuario al registre deberá especificar su posición en la escuela (alumno, administrativo, etc….)
* Cada libro, tesis, revista, usuario se le dará un ID para identificarlo.
* De existir se deberá pagar una penalización por retorno tardío.

Use case spcification

usuario

administradores

Diagrama de caso de uso de librería

Use case: registro

Actores principales: administrador

Descripción: el administrador hará el registro de los nuevos usuarios

Precondición: el usuario deberá tener una id con el cual se le identificará

Flujo de los eventos:

* El administrador deberá de crear una cuenta nueva
* Si ya está registrado no es necesario crear la cuenta nueva
* El administrador creara una nueva cuenta para el usuario

Escenarios alternos:

* Error al hacer login

Pos condición:

* El sistema ya está con una sesión iniciada

usuario

administrador

Use case: login

Actores principales: administrador, usuarios

Descripción: el administrado o usuario hará login e la pagina

Precondición: el usuario deberá una cuenta previamente creada

Flujo de los eventos:

* El administrador o usuario deberá ingresar a la página de login
* Se deberá ingresar los datos correctos para hacer login
* Si los datos son correctos se mostrará la página principal

Escenarios alternos:

* Error al hacer login

Postcondición:

* El sistema ya está con una sesión iniciada

administrador

administrador

Use case: libro(añadir, eliminar)

Actores principales: administrador

Descripción: el administrado o usuario hara login e la pagina

Flujo de los eventos:

* El administrador o usuario deberá ingresar a la pagina de login
* Se deberá ingresar los datos correctos para hacer login
* Si los datos son correctos se mostrara la pagina principal y después a registro de libros
* Si el libro no ha sido registrado antes, entonces puede ser añadido a la base de datos
* Si se desea eliminar un libro se tiene que hacer una búsqueda para encontrarlo y eliminarlo

Escenarios alternos:

* Error al hacer login
* Si un libro ya esta registrado no debería de ingresarse de nuevo
* Si el libro no ha sido registrado no puede ser eliminado

Postcondición:

* Se modifica la base de datos y los registros del libro

administrador

Use case: prestamo

Actores principales: administrador

Descripción: el usuario pedirá préstamo de un libro

Flujo de los eventos:

* El administrador o usuario deberá ingresar a la pagina de login
* Se deberá ingresar los datos correctos para hacer login
* Si los datos son correctos se mostrara la pagina principal.
* Sxe buscara el libro deseado
* Si se encuentra el usuario solicitara el libro al administrador
* El administrador registrara el préstamo al usuario indicado

Escenarios alternos:

* Error al hacer login
* Libro no encontrado

Postcondición:

* Se modifica la base de datos y los registros del libro

administrador

usuario

Software Architecture Document

# Introduction

## Purpose

Este documento proporciona una vista general de la arquitectura del Sistema, usando diferentes vistas de arquitectura para mostrar diferentes aspectos del Sistema.

arquitectura de software se refiere a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. Esta estructuración representa un diseño de alto nivel del sistema que tiene dos propósitos primarios: satisfacer los atributos de calidad (desempeño, seguridad, modificabilidad), y servir como guía en el desarrollo. Al igual que en la ingeniería civil, las decisiones críticas relativas al diseño general de un sistema de software complejo deben de hacerse desde un principio.

El propósito del SAD es especificar de forma clara la Arquitectura de Software que se utilizará para el desarrollo y la implementación del Sistema Ide biblioteca. Este documento se orienta para la lectura por parte de los Stakeholders implicados en el proceso de desarrollo de la aplicación, por consiguiente, se convertirá en la principal fuente de información para iniciar el desarrollo de la aplicación y los componentes necesarios para el correcto funcionamiento de la aplicación.

# Architectural Representation

Vista de Casos de Uso: lista los casos de uso o escenarios del modelo de casos de uso que representen funcionalidades centrales del sistema final, que requieran una gran cobertura arquitectónica o aquellos que impliquen algún punto especialmente delicado de la arquitectura.

• Vista Lógica: describe las partes arquitectónicamente significativas del modelo de diseño, como ser la descomposición en capas, subsistemas o paquetes. Una vez presentadas estas unidades lógicas principales, se profundiza en ellas hasta el nivel que se considere adecuado.

• Vista de Procesos: describe la descomposición del sistema en threads y procesos pesados. Indica que procesos o grupos de procesos se comunican o interactúan entres sí y los modos en que estos se comunican.

• Vista de Deployment: describe uno o más escenarios de distribución física del sistema sobre los cuales se ejecutará y hará el deploy del mismo. Muestra la comunicación entre los diferentes nodos que componen los escenarios antes mencionados, así como el mapeo de los elementos de la Vista de Procesos en dichos nodos.

• Vista de Implementación: describe la estructura general del Modelo de Implementación y el mapeo de los subsistemas, paquetes y clases de la Vista Lógica a subsistemas y componentes de implementación. • Vista de Datos: describe los elementos principales del Modelo de Datos, brindando un panorama general de dicho modelo en términos de tablas, vistas, índices, etc.

# Architectural Goals and Constraints

El desarrollo de la aplicación se enfoca en que llegue a tener características que sean sostenibles, eficientes y fáciles de usar para cualquier usuario del sistema biblioteca

Nuestras principales metas a nivel de arquitectura son las siguientes:

* Performance: El desempeño de la aplicación debe ser eficiente de tal manera que el usuario pueda usar de manera rápida y sencilla el sistema
* Usabilidad: El diseño debe ser orientado por y para la comodidad del usuario, de manera que la interfaz sea intuitiva y fácil de manejar, al mismo tiempo que se fomente altamente la interacción entre ambos.

En la planificación de este proceso hemos encontrado las siguientes restricciones:

1. Restricciones de contenido: debido a que nuestro sistema está basado y propuesto para aquellos que pertenecen a laU de G, el mismo debe estar de acuerdo con el reglamento de la institución, por lo que ciertos contenidos deben ser excluidos en el uso de la aplicación. .

# Use-Case View

Ver seccion de espicifacion de casos de uso

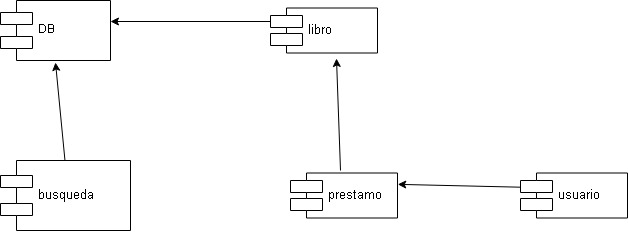
# Logical View

Se presentan en este punto los sucesivos refinamientos que definen las diferentes unidades lógicas que componen la arquitectura de la biblioteca.

El primer refinamiento realizado consiste en la descomposición en subsistemas. Los subsistemas representan cortes verticales al diseño del sistema.

Cada subsistema consiste en el agrupamiento de diferentes funcionalidades relacionadas entre sí y posee la capacidad de funcionar como un sistema en sí mismo.

Para proporcionar una base para comprender la estructura y la organización del diseño del sistema, se utiliza una vista de la arquitectura llamada Vista lógica en el flujo de trabajo de análisis y diseño. Sólo hay una vista lógica del sistema, que ilustra las realizaciones de guiones de uso clave, subsistemas, paquetes y clases que abarcan el comportamiento significativo arquitectónicamente. La vista lógica se perfecciona durante las iteraciones



Los diagramas de clases son uno de los tipos de diagramas más útiles en UML, ya que trazan claramente la estructura de un sistema concreto al modelar sus clases, atributos, operaciones y relaciones entre objetos.

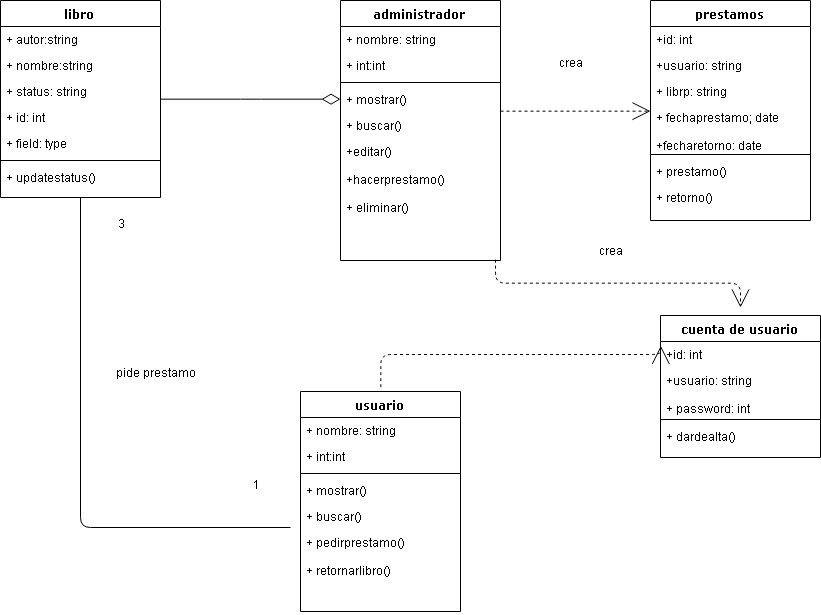
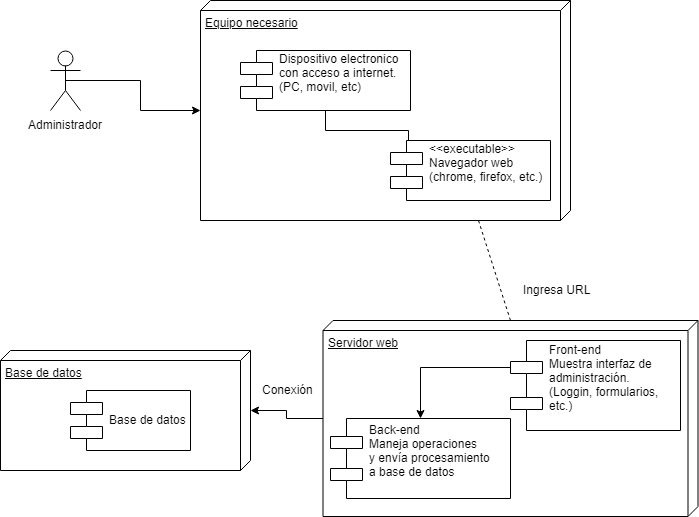


Diagrama de clases

# Deployment View

Para proporcionar una base para comprender la distribución física del sistema en un conjunto de nodos de proceso, se utiliza una vista de la arquitectura llamada vista de despliegue en el flujo de trabajo de análisis y diseño. La vista de despliegue (una de las cinco vistas, que se muestran a continuación) ilustra la distribución de procesos en un conjunto de nodos del sistema, incluida la distribución física de procesos y hebras. La vista de despliegue se perfecciona durante las iteraciones.

A continuación, se muestra un diagrama de despliegue que modela, a alto nivel, la distribución de las piezas de software del sistema biblioteca, sobre los elementos de hardware que se usarán para ejecutarla, indicando las asociaciones entre nodos con caminos de comunicación, que indican la tecnología requerida para que ésta se lleve a cabo exitosamente.



# Size and Performance

La aplicación no demanda una gran cantidad de espacio, sin embargo, se tiene que tomar en cuenta la cantidad de usuarios que van a hacer uso de ella.

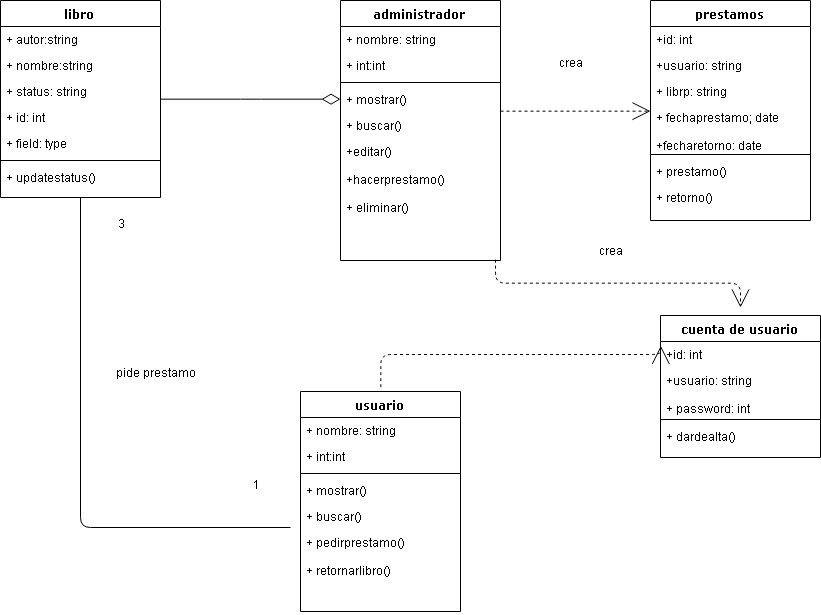
Podría requerirse un máximo de entradas en el Cuaderno de tal manera que se tenga un límite de almacenamiento.

Fase de elaboración

**Diagrama de clases**

El diagrama de clases recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos, pero no muestra información temporal.

Con el fin de facilitar la comprensión del diagrama, se pueden incluir paquetes como elementos del mismo, donde cada uno de ellos agrupa un conjunto de clases.



**Modelo de diseño(diagramas de actividad)**

Diagrama de actividad para la generación de un préstamo

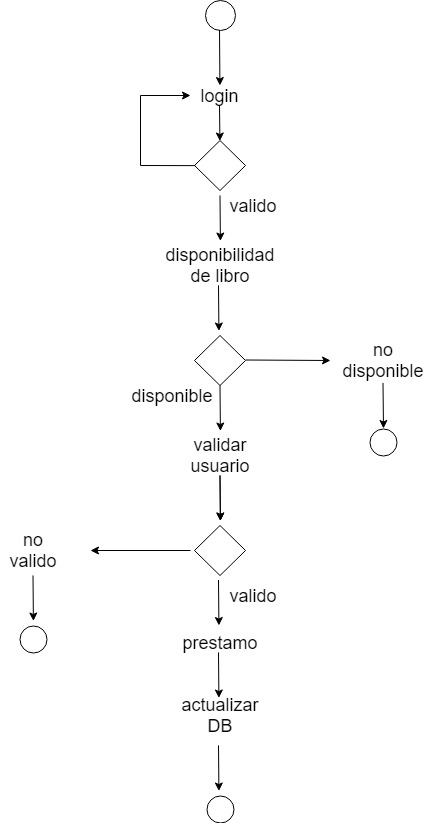


Diagrama de actividad para el registro de un usuario

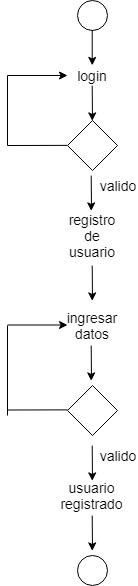
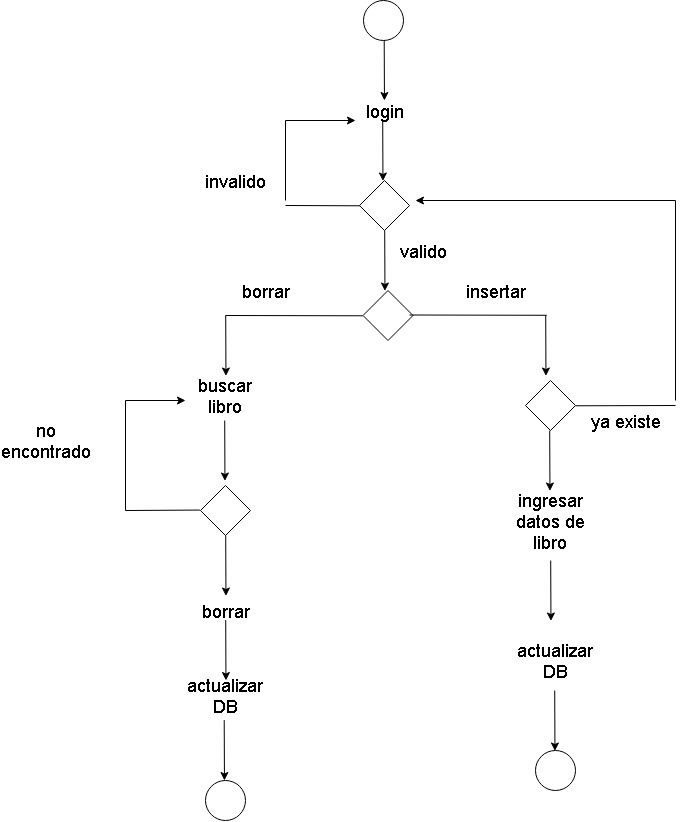
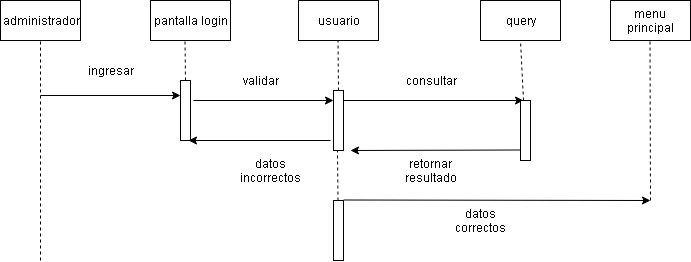


Diagrama para eliminación de un libro y registro de un libro



Diagramas de secuencia

Registro de usuario



Registrar libro

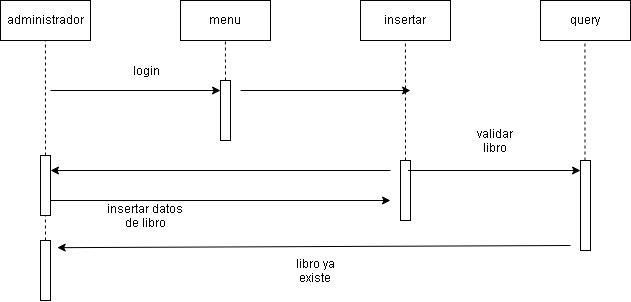
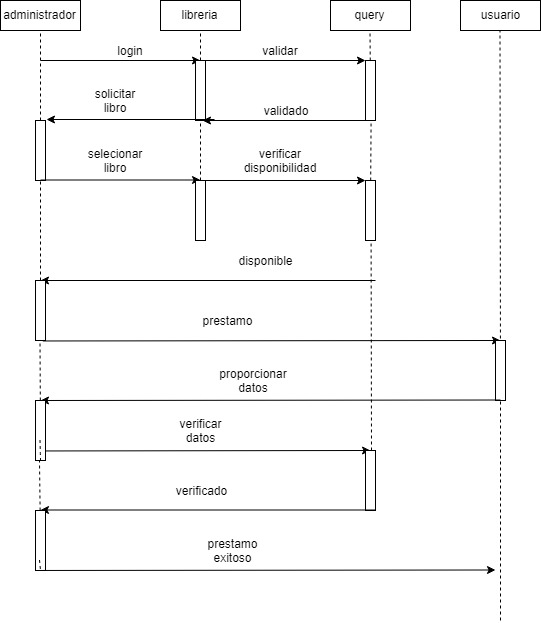
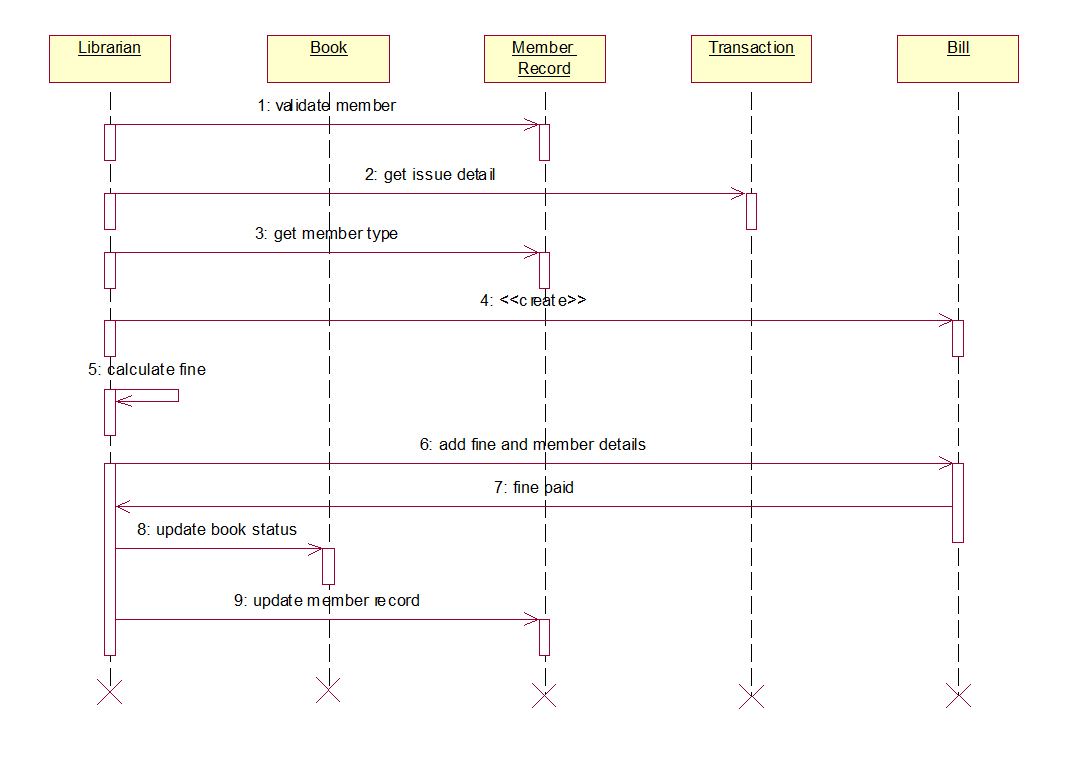
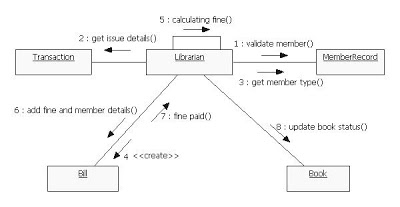


Diagrama de prestamo





Diagramas de colaboracion



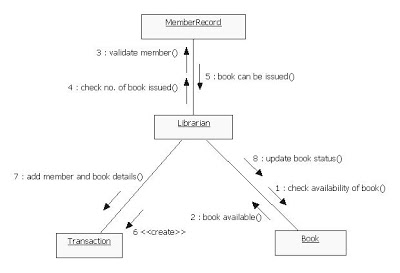


Diagram de components

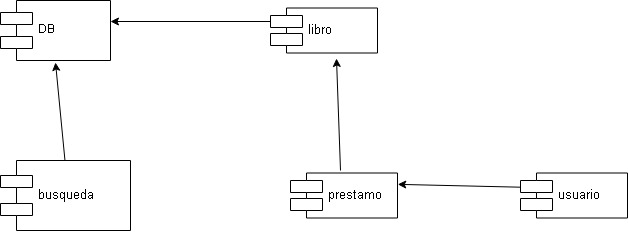


Diagrama de despliegue

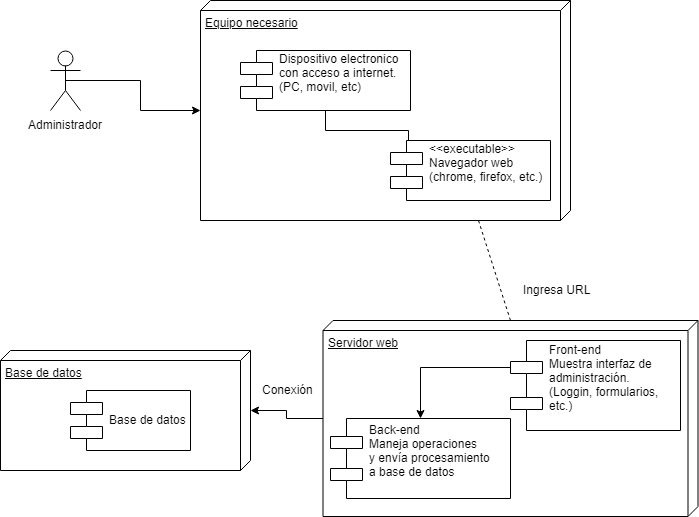
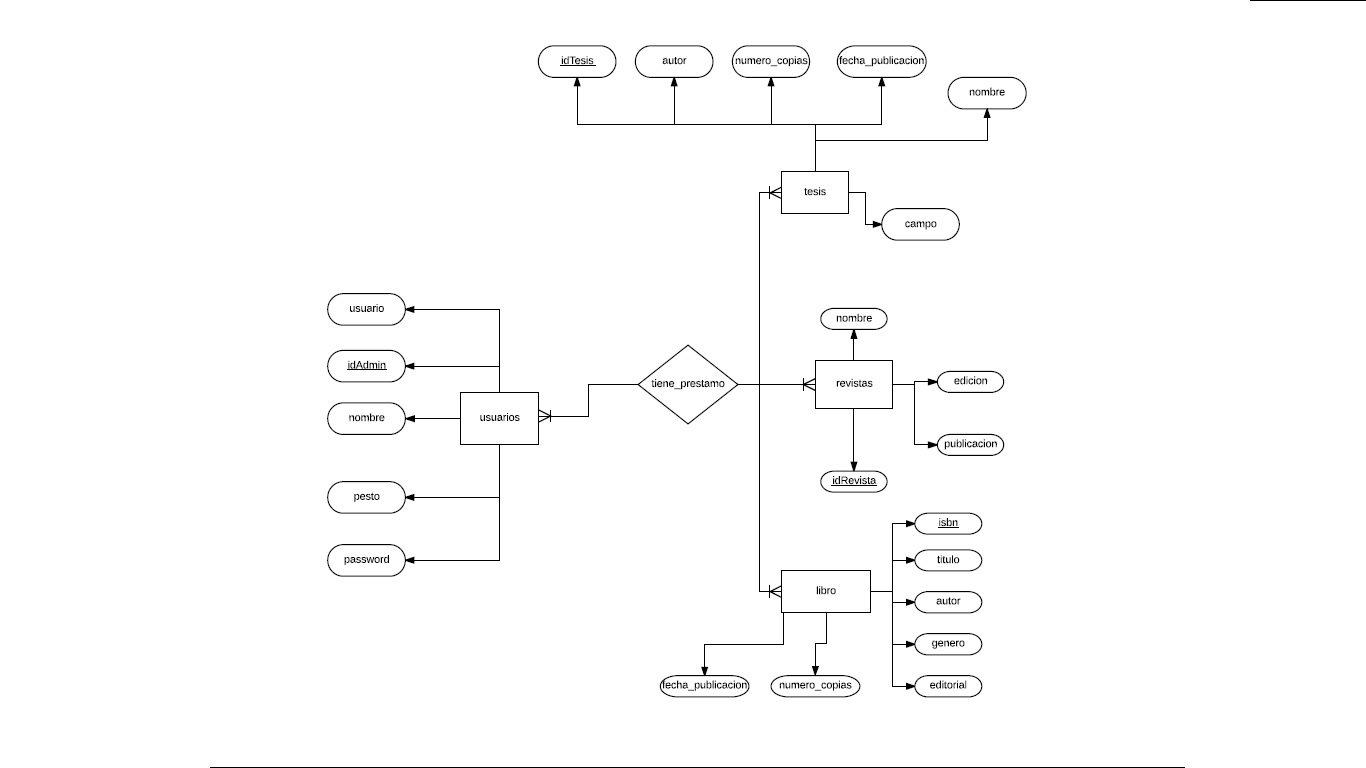


Diagram ER



<Iteration/ Master> Test Plan

# Introduction

## Purpose

El proposito de el plan de test es coleccionar toda la informacion necesaria para planear y controlar las pruebas para cada iteración . describe el acercamiento al testeo del software.

El plan de pruebas para el sistema de bibliotecaapoya los siguientes objetivos:

* Identificar las partes que deben de ser puestas a prueba
* Identificar los recursos y proporcionar un estimado para las pruebas
* Listar las pruebas a realizar

## Scope

En el Proyecto se haran dierentes tipo de pruebas:

Pruebas unitarias: Son principalmente trozos de código diseñados para comprobar que el código principal está funcionando como esperábamos. Pequeños test creados específicamente para cubrir todos los requisitos del código y verificar sus resultados.

### Pruebas de integración

Las pruebas de integración verifican que los distintos módulos o servicios utilizados por tu aplicación funcionan bien en conjunto. Por ejemplo, se puede probar la interacción con la base de datos o asegurarse de que los microservicios funcionan bien en conjunto y según lo esperado. Estos tipos de pruebas son más costosos de ejecutar, ya que requieren que varias partes de la aplicación estén en marcha.

### Pruebas estáticas

Son el tipo de pruebas que se realizan sin ejecutar el código de la aplicación.

Puede referirse a la revisión de documentos, ya que no se hace una ejecución de código. Esto se debe a que se pueden realizar "pruebas de escritorio" con el objetivo de seguir los flujos de la aplicación.

# Evaluation Mission and Test Motivation

## Background

En un proyecto de desarrollo de software pueden aparecer errores en cualquiera de las etapas del ciclo de vida, algunos de ellos incluso permanecen sin ser descubiertos, de ahí la importancia de las pruebas en desarrollo de software.

Hay una gran probabilidad de que el código final tenga errores tanto de requerimientos, como de diseño o de funcionalidad. Para identificar estos problemas antes de que ocurran en un entorno crítico, es necesario realizar pruebas de software, una parte muy importante del proceso.

Al realizar el proyecto de la biblioteca se llevaran a cabo estas pruebas en el transcurso de la fase de desarrollo.

## Evaluation Mission

[Provide a brief statement that defines the mission for the evaluation effort in the current iteration. This statement might incorporate one or more concerns including:

* find as many bugs as possible
* find important problems, assess perceived quality risks
* advise about perceived project risks
* certify to a standard
* verify a specification (requirements, design or claims)
* advise about product quality, satisfy stakeholders
* advise about testing
* fulfill process mandates
* and so forth

Each mission provides a different context to the test effort and alters the way in which testing should be approached.]

De este modo, los objetivos principales al realizar pruebas serán los siguientes:

* Detectar errores en el código
* Detectar errores específicos
* Reducir costos de mantenimiento
* Aportar a que el producto sea de mayor calidad
* Detectar posibles riesgos en el proyecto
* Apoyar el proceso de calidad
* Solucionar problemas antes de finalizar el proyecto
* Verificar que el software hace lo que se pide

## Test Motivators

[Provide an outline of the key elements that will motivate the testing effort in this iteration. Testing will be motivated by many things⎯quality risks, technical risks, project risks, use cases, functional requirements, non-functional requirements, design elements, suspected failures or faults, change requests, and so forth.]

# Target Test Items

Partes a las que se le aplicaran las pruebas:

* Base de datos
* Código mientras este se esté desarrollando
* Hardware en el que se instalara el sistema
* Interfaz que usaran los usuarios
* Interfaz que usara el administrador

# Test Approach

Esta seccion describe como las pruebas seran realizadas, un aspect a considerer para las pruebas son las tecnicas que seran utilizadas, se debera incluir como cada tecnica puede ser implementada (manualmente o automatizada) .

Por cada técnica se deberá inclir una descripción de esta y definir porque es una parte importante de el testeo del proyecto.

## Testing Techniques and Types

### Data and Database Integrity Testing

Pruebas de integridad de base de datos son pruebas de los métodos y procesos utilizados para acceder y gestionar datos(base de datos), para asegurar que los métodos de acceso, los procesos y las reglas de los datos funcionan como se espera y que durante el acceso a la base de datos, los datos no se corrompan, sean borrados, modificados o creados de forma inesperada.

|  |  |
| --- | --- |
| Technique Objective: | Pruebas de integridad de base de datos son pruebas de los métodos y procesos utilizados para acceder y gestionar datos(base de datos), para asegurar que los métodos de acceso, los procesos y las reglas de los datos funcionan como se espera y que durante el acceso a la base de datos, los datos no se corrompan, sean borrados, modificados o creados de forma inesperada. |
| Technique: | * Probar los metodos de acceso a la DB enviando dtoas validos e inavlidos paraverificar su funcionalidad * Inspeccionar que la DB para asegurar que la información se ha guardado de la manera que se debía, asi como verificar que los procesos fueron realizados correctamente. * Los resultados esperados ocurren cuando se utiliza un dato válido.   Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido. |
| Oracles: | El objetivo de estas pruebas es verificar el ingreso, procesamiento y recuperación apropiado de datos, y la implementación apropiada de las reglas de negocios. Este tipo de pruebas se basan en técnicas de caja negra, ésto es, verificar el sistema (y sus procesos internos), la interacción con las aplicaciones que lo usan via GUI y analizar las salidas o resultados. |
| Required Tools: | * Test Script Automation Tool * base configuration imager and restorer * backup and recovery tools * installation-monitoring tools (registry, hard disk, CPU, memory, and so forth) * database SQL utilities and tools * Data-generation tools] |
| Success Criteria: | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identifcaron han sido tenidos en cuenta. |

### Function Testing

[Function testing of the target-of-test should focus on any requirements for test that can be traced directly to use cases or business functions and business rules. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box techniques; that is verifying the application and its internal processes by interacting with the application via the Graphical User Interface (GUI) and analyzing the output or results. The following table identifies an outline of the testing recommended for each application.]

|  |  |
| --- | --- |
| Technique Objective: | Las pruebas de desempeño miden tiempos de respuesta, índices de procesamiento de transacciones y otros requisitos sensibles al tiempo. El objetivo de las pruebas de desempeño es verificar y validar los requisitos de desempeño que se han especificado |
| Technique: | * Las pruebas de desempeño usualmente se ejecutan varias veces, utilizando en cada una, carga diferente en el sistema. La prueba inicial debe ser ejecutada con una carga similar a la esperada en el sistema. Una segunda prueba debe hacerse utilizando una carga máxima. * Adicionalmente, las pruebas de desempeño pueden ser utilizadas para perfilar y refinar el desempeño del sistema como una función de condiciones tales como carga o configuraciones de hardware |
| Oracles: | * Utilizar los procedimientos de prueba desarrollados para las pruebas del modelo del negocio (Pruebas del Sistema).· * Modificar archivos de datos (para incrementar el número de transacciones) o los scripts para incrementar el número de veces que ocurre cada transacción.· * Los scripts deben ser ejecutados en una máquina y deben ser repetidos con múltiples clientes (virtuales o actuales). |
| Required Tools: | [The technique requires the following tools:   * Test Script Automation Tool * base configuration imager and restorer * backup and recovery tools * installation-monitoring tools (registry, hard disk, CPU, memory, and so forth) * Data-generation tools] |
| Success Criteria: | * Un Usuario / Una Transaccion. Se completaron las pruebas sin ninguna falla y dentro del tiempo esperado o requerido por transacción. * Múltiples transacciones, múltiples usuarios. Se completaron las pruebas de los scripts sin ninguna falla y dentro del tiempo esperado. |

### Business Cycle Testing

[Business Cycle Testing should emulate the activities performed on the <Project Name> over time. A period should be identified, such as one year, and transactions and activities that would occur during a year’s period should be executed. This includes all daily, weekly, and monthly cycles, and events that are date-sensitive, such as ticklers.]

|  |  |
| --- | --- |
| Technique Objective: | Asegurar que el sistema funciona de acuerdo con el modelo de negocios emulando todos los eventos en el tiempo y en función del tiempo. |
| Technique: | ncremente el número de veces en que una función es ejecutada para simular diferentes usuarios sobre un periodo especificado  Todas las fechas o funciones que involucren tiempos serán probadas con datos válidos e inválidos de fechas o periodos de tiempo.  Todas las funciones ocurren en un periodo de tiempo serán ejecutadas en el tiempo apropiado. |
| Oracles: | Las pruebas del ciclo de negocio deberían emular las actividades ejecutadas en el a través del tiempo.  Debería identificarse un periodo, como por ejemplo un año, y las transacciones y actividades que podrían ocurrir durante un periodo de un año deberían ejecutarse. Incluyendo todos los ciclos y eventos diarios, semanales y mensuales que sean datos sensitivos, como las agendas. |
| Required Tools: | [The technique requires the following tools:   * Test Script Automation Tool * base configuration imager and restorer * backup and recovery tools * Data-generation tools] |
| Success Criteria: | Todas las pruebas planeadas han sido ejecutadas.  Todos los defectos que se identificaron han sido tenidos en cuenta. |

### User Interface Testing

[User Interface (UI) testing verifies a user’s interaction with the software. The goal of UI testing is to ensure that the UI provides the user with the appropriate access and navigation through the functions of the target-of-test. In addition, UI testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.]

|  |  |
| --- | --- |
| Technique Objective: | Verifica lo siguiente:   * La navegación a través de los objetos de la prueba reflejan las funcionalidades del negocio y requisitos, se realiza una navegación ventana por ventana, usando los modos de acceso (tabuladores, movimientos del mouse, teclas rápidas, etc) * Los objetos de la ventana y características, tales como menús, medidas, posiciones, estados y focos se verifican conforme a los estándares. |
| Technique: | Pruebas de crear / modificar cada ventana para verificar la adecuada navegación y estado de los objetos |
| Oracles: | La prueba de interfaz de usuario verifica la interacción del usuario con el software. El objetivo es asegurar que la interfaz tiene apropiada navegación a través de las diferentes funcionalidades. Adicionalmente, las pruebas de interfaz aseguran que los objetos de la interfaz a ser probada se encuentra dentro de los estandares de la industria |
| Required Tools: | [The technique requires the Test Script Automation Tool.] |
| Success Criteria: | Cada ventana elegida será totalmente verificada y comparada con similares en el mercado logrando una buena aceptación dentro del estándar. |

### Stress Testing

[Stress testing is a type of performance test implemented and executed to understand how a system fails due to conditions at the boundary, or outside of, the expected tolerances. This typically involves low resources or competition for resources. Low resource conditions reveal how the target-of-test fails that is not apparent under normal conditions. Other defects might result from competition for shared resources, like database locks or network bandwidth, although some of these tests are usually addressed under functional and load testing.]

[Note: References to transactions in the following table refer to logical business transactions.]

|  |  |
| --- | --- |
| Technique Objective: | Verificar que el sistema funciona apropiadamente y sin errores, bajo estas condiciones de stress:   * Memoria baja o no disponible en el servidor. * Máximo número de clientes conectados o simulados (actuales o físicamente posibles) * Múltiples usuarios desempeñando la misma transacción con los mismos datos. * El peor caso de volumen de transacciones (ver pruebas de desempeño). |
| Technique: | * Use los scripts utilizados en las pruebas de desempeño. * Para probar recursos limitados, las pruebas se deben correr en un servidor con configuración reducida (o limitada). * Para las pruebas de stress restantes, deben utilizarse múltiples clientes, ya sea corriendo los mismos scripts o scripts complementarios para producir el peor caso de volumen de transacciones. |
| Oracles: | Las pruebas de stress se proponen encontrar errores debidos a recursos bajos o completitud de recursos. Poca memoria o espacio en disco puede revelar defectos en el sistema que no son aparentes bajo condiciones normales.  Otros defectos pueden resultar de incluir recursos compartidos, como bloqueos de base de datos o ancho de banda de la red.  Las pruebas de stress identifican la carga máxima que el sistema puede manejar. |
| Required Tools: | [The technique requires the following tools:   * Test Script Automation Tool * Transaction Load Scheduling and control tool * installation-monitoring tools (registry, hard disk, CPU, memory, and so on) * resource-constraining tools (for example, Canned Heat) * Data-generation tools] |
| Success Criteria: | Todas las pruebas planeadas han sido ejecutadas y excedidas sin que el sistema falle. ( O si las condiciones en que el sistema falle ocurren por fuera de las condiciones especificadas). |

### 

### Security and Access Control Testing

|  |  |
| --- | --- |
| Technique Objective: | * Nivel de seguridad de la aplicación: Verifica que un actor solo pueda acceder a las funciones y datos que su usuario tiene permitido. * Nivel de Seguridad del Sistema: Verificar que solo los actores con acceso al sistema y a la aplicación están habilitados para accederla. |
| Technique: | * Funciones / Seguridad de Datos: Identificar cada tipo de usuario y las funciones y datos a los que se debe autorizar. * Crear pruebas para cada tipo de usuario y verificar cada permiso, creando transacciones específicas para cada tipo de usuario. * Modificar tipos de usuarios y volver a ejecutar las pruebas. En cada caso, verificar si los datos o funciones adicionales quedan correctamente permitidos o denegados. |
| Oracles: | Las pruebas de seguridad de la aplicación garantizan que, con base en la seguridad deseada, los usuarios están restringidos a funciones específicas o su acceso está limitado únicamente a los datos que está autorizado a acceder.  Las pruebas de seguridad y control de acceso se centran en dos áreas claves de seguridad:  Seguridad del sistema, incluyendo acceso a datos o Funciones de negocios y  Seguridad del sistema, incluyendo ingresos y accesos remotos al sistema. |
| Required Tools: | [The technique requires the following tools:   * Test Script Automation Tool * “Hacker” security breach and probing tools * OS Security Admin Tools] |
| Success Criteria: | Para cada tipo de usuario conocido, las funciones y datos apropiados y todas las transacciones funcionan como se esperaba. |

### Configuration Testing

|  |  |
| --- | --- |
| Technique Objective: | Validar y verificar que el cliente del sistema funciona apropiadamente en las estaciones de trabajo recomendadas. |
| Technique: | Use los scripts para Integración y Pruebas del Sistema.  Incluya la apertura o cierre de varias aplicaciones Microsoft, como Excel y Word (o algun tipo de software similar a la que se esta probando ) como una parte de la prueba, ya sea al comienzo o en algún momento intermedio.  Ejecute algunas transacciones para simular actividades cotidianas del usuario, dentro y fuera de las aplicaciones que interactúan con la Base. |
| Oracles: | Estas pruebas verifican la operación del sistema en diferentes configuraciones de hardware y software.  En la mayoría de los ambientes de producción, las especificaciones para las estaciones de trabajo, equipos de red y servidores pueden variar. Las estaciones pueden tener diferentes versiones de software instaladas (Sistemas Operativos, Drivers, etc) y en cualquier momento, pueden llegar a utilizarse diferentes combinaciones. |
| Required Tools: | [The technique requires the following tools:   * base configuration imager and restore * installation monitoring tools (registry, hard disk, CPU, memory, and so on)] |
| Success Criteria: | Para cada combinación de aplicaciones que interactúan con la Base de datos a probar, las transacciones deben ser ejecutadas sin fallas. |

### Installation Testing

|  |  |
| --- | --- |
| Technique Objective: | Verificar y validar que el sistema se instala apropiadamente en cada cliente, bajo las siguientes condiciones:   * Instalaciones nuevas, nuevas máquinas a las que nunca se les ha instalado el sistema. * Actualizar máquinas previamente instaladas con el sistema. * Instalar versiones viejas en máquinas previamente instaladas con el sistema. |
| Technique: | Diseñar sripts para validar las condiciones de la máquina a instalar .  Realizar la instalación |
| Oracles: | Las pruebas de instalación tienen dos propósitos. El primero es asegurar que el sistema puede ser instalado en todas las configuraciones posibles, tales como nuevas instalaciones, actualizaciones, instalaciones completas o personalizadas, y bajo condiciones normales o anormales; estas últimas incluyen insuficiente espacio en disco, falta de privilegios para algunas tareas, etc.  El segundo propósito es verificar que, una vez instalado, el sistema opera correctamente. Esto usualmente implica correr un número significativo de pruebas de Funcionalidad. |
| Required Tools: | [The technique requires the following tools:   * base configuration imager and restorer * installation monitoring tools (registry, hard disk, CPU, memory, and so on)] |
| Success Criteria: | Las transacciones de la aplicación se ejecutan sin fallas |

# 

# Entry and Exit Criteria

## Test Plan

### Test Plan Entry Criteria

[Specify the criteria that will be used to determine whether the execution of the Test Plan can begin.]

Se iniciara el plan de pruebas cando se haya diseñando y planeado la base de datos y la interfaz, asi como tener la primera parte del codigo desarrollado.

### Test Plan Exit Criteria

Se dará por finalizado el plan de pruebas cuando se considere que el proyecto ha cumplido todos los requerimientos y objetivos que se plantearon.

### Suspension and Resumption Criteria

Se suspenderá el plan de pruebas si se observa que el proceso que se esta llevando a cabo no es el adecuado y se tendrá que replantear el proceso a seguir

# Iteration Milestones

| Milestone | Planned Start Date | Actual Start Date | Planned End Date | Actual End Date |
| --- | --- | --- | --- | --- |
| Iteration Plan agreed | 10 feb 2019 | 10 feb 2019 | 15 feb 2019 | 17 feb 2019 |
| Iteration starts | 19 feb 2019 | 20 feb 2019 | 2 mar 2019 | 2 mar 2019 |
| Requirements baselined | 4 mar 2019 | 4 mar 2019 | 10 mar 2019 | 10 mar 2019 |
| Architecture baselined | 13 mar 2019 | 15 mar 2019 | 22 mar 2019 | 22 mar 2019 |
| User Interface baselined | 22 mar 2019 | 22 mar 2019 | 29 mar 2019 | 29 mar 2019 |
| First Build delivered to test | 1 abri 2019 | 1 abri 2019 | 9 abri 2019 | 11 abri 2019 |
| First Build accepted into test | 11 abri 2019 | 12 abri 2019 | 20 abri 2019 | 22 abri 2019 |
| Second Build delivered to test | 25 arbi 2019 | 25 abri 2019 | 29 abri 2019 | 30 abri 2019 |
| Second Build accepted into test | 1 may 2019 | 2 may 2019 | 10 may 2019 | 10 may 2019 |
| Third Build delivered to test | 10 may 2019 | 10 may 2019 | 15 may 2019 | 15 may 2019 |
| Third Build accepted into test | 15 may 2019 | 15 may 2019 | 19 may 2019 | 20 may 2019 |
| Iteration Assessment review | 20 may 2019 | 20 may 2019 | 23 may 2019 | 23 may 2019 |
| Iteration ends |  |  | 25 may 2019 | 25 may 2019 |

Software development plan

Project Overview

## Project Purpose, Scope, and Objectives

Este proyecto busca atacar este problema por medio del desarrollo de un Sistema controlar y almacenar la información sobre los libros, asi como los préstamos que se hicieron y las sanciones por entrega tardía.

Cada libro guardado en la biblioteca cuenta con un título, un autor, su editorial, ISBN, la fecha de publicación y el género del libro o a que rama universitaria pertenece. Aparte se mantiene un registro del número de copias en existencia actualmente en la biblioteca para llevar un control de los préstamos y revisar si un libro se puede prestar o no.

Por otra parte, también mantiene un registro de sus empleados en todas las áreas de la biblioteca, cada empleado cuenta con Nombre, Edad, Puesto que desempeña y una ID de empleado. Un empleado tiene la oportunidad de sacar libros de la biblioteca por lo que se utiliza su ID al momento de hacer el préstamo.

# Project Organization

## Organizational Structure

El encargado del proyecto y el sistema solo es una persona no hay jerarquía de ningún tipo

## Roles and Responsibilities

El encargado principal se encargara de todo lo relacionado al proyecto

# Management Process

## Project Estimates

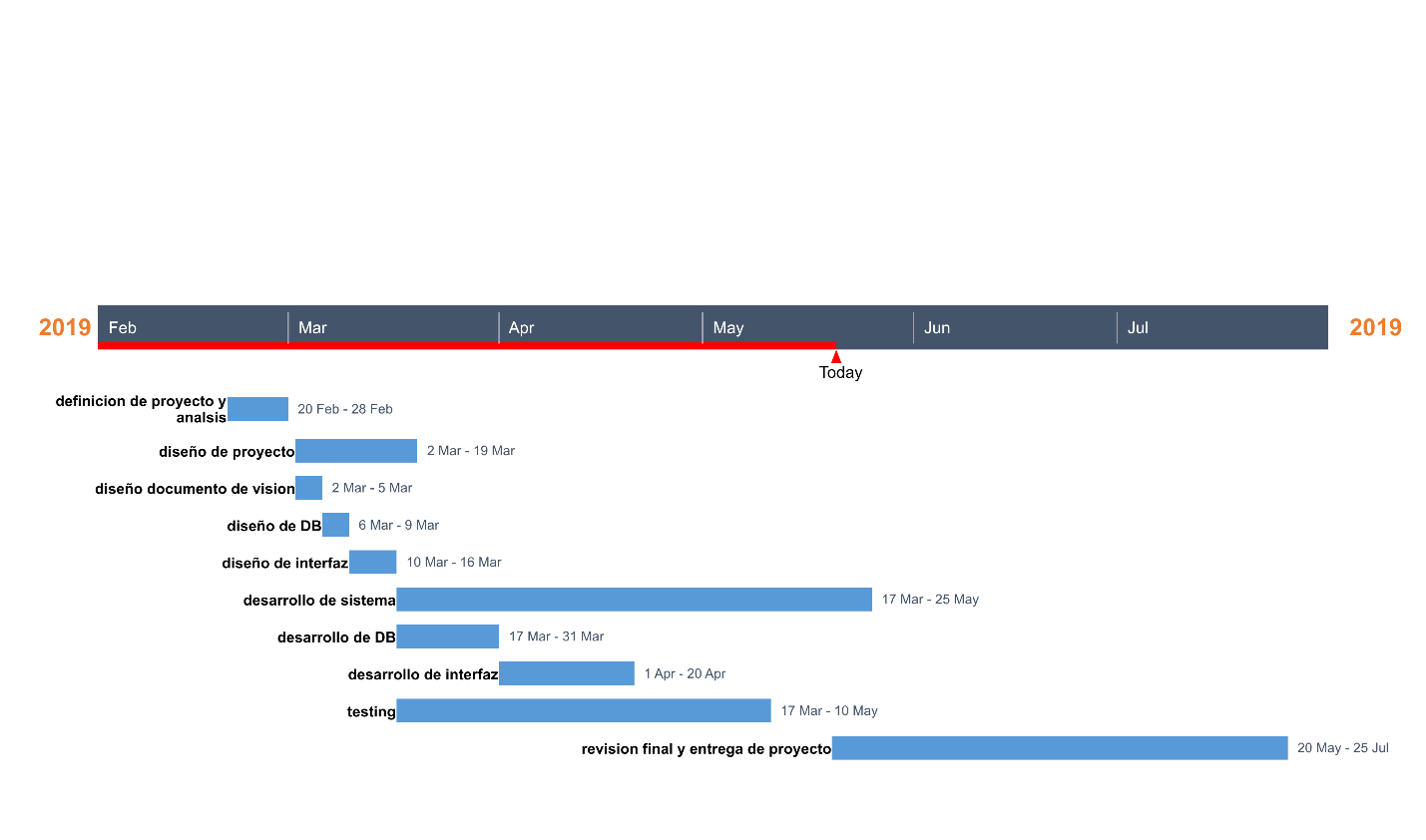
.no se espera ningún gasto en la realización del Proyecto, el tiempo de entrega se espera sea de 3 a 5 meses,

## Project Plan

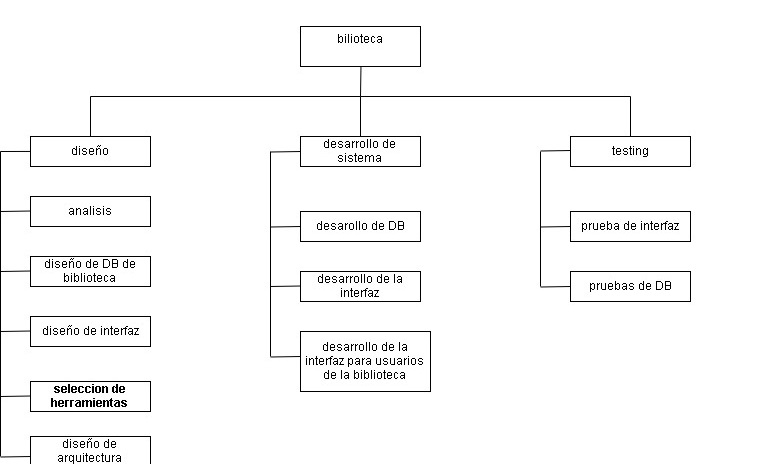
### Phase Plan

[Include the following:

* Work Breakdown Structure (WBS)



* a timeline or Gantt chart showing the allocation of time to the project phases or iterations



### Releases

[A brief description of each software release and whether it’s demo, beta, and so on.]

### Project Schedule

Calendario del proyecto

|  |  |  |
| --- | --- | --- |
| Actividad# | Descripción de actividad | Semanas |
| 1 | Initial System requirement | 2 |
| 2 | Requirement specifiction | 2 |
| 3 | Desing specification | 3 |
| 4 | Data base creation | 2 |
| 5 | Software development | 7 |
| 6 | Software testing | 1 |
| 7 | Software documentation | 1 |
| 8 | Software installation | 1 |
| 9 | Total estimated duration | 19 |
|  | Assumption: equipo de un ingeniero |  |

## **Project Monitoring and Control**

### Schedule Control Plan

[Describe the approach taken to monitor progress against the planned schedule and how to take corrective action when required.]

Para monitorear el progreso y poder reaccionar ante posibles fallas en el proceso se tendra un itineario el cual se debera seguir lo mas cerca possible, si se llega a presentar alguna incoveniencia e el proceso se debera reevaluar para saber la mejor manera de proceder

### Quality Control Plan

Si se cuenta con un **plan de control de calidad** sólido, tiene la ventaja de poder identificar las causas que dan lugar a problemas de calidad y, desde ese conocimiento, trabajar para garantizar que los entregables el proyecto y las condiciones de trabajo en general cumplen con los requisitos de calidad acordados.

Para elaborar un buen plan de control de calidad es recomendable apoyarse en los siguientes elementos:

* Plan de gestión de calidad.
* Plan de mejora continua.
* Métricas de calidad.
* Checklist de calidad.

Actualizaciones de la documentación de proyecto

A partir de estos elementos será posible determinar todo lo que se necesita para un plan de control de calidad efectivo.

Comprendiendo esto se puede actuar ante problrmas que se presenten durante el desarrollo del proyecto

Manual

Manual

Introducción

El plantel de CUCEI (Centro Universitario de Ciencias Exactas e Ingenierías) de la Universidad de Guadalajara requiere de un nuevo sistema para su biblioteca.

La biblioteca tiene clasificado a todo en cinco divisiones:

* LIBROS
* EMPLEADOS
* CLIENTES
* TESIS
* SANCIONES

.

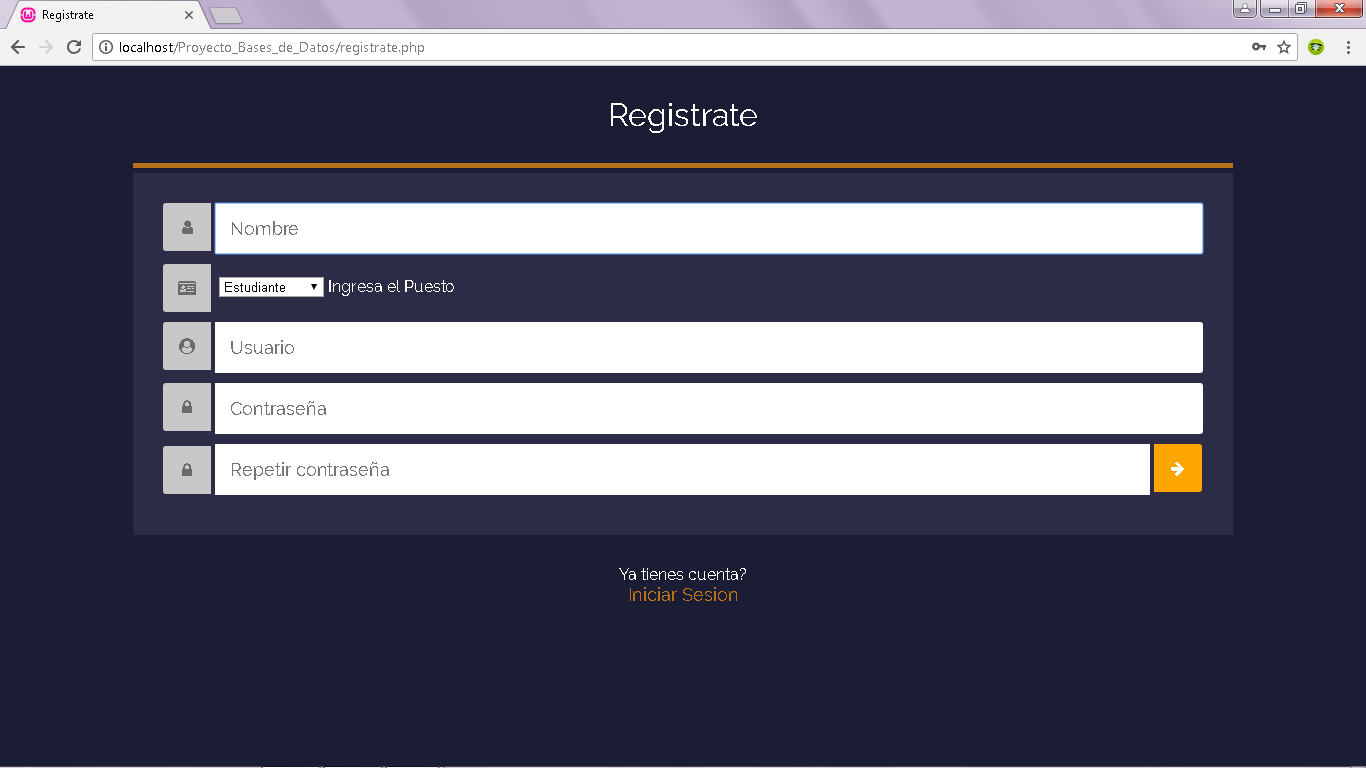
Si la biblioteca no lleva un buen manejo de los prestamos puede ocasionar perdida de libros, tesis, etc.., así como confusión de los usuarios de la biblioteca.

Este proyecto busca atacar este problema por medio del desarrollo de un Sistema controlar y almacenar la información sobre los libros, así como los préstamos que se hicieron y las sanciones por entrega tardía.

Cada libro guardado en la biblioteca cuenta con un título, un autor, su editorial, ISBN, la fecha de publicación y el género del libro o a que rama universitaria pertenece. Aparte se mantiene un registro del número de copias en existencia actualmente en la biblioteca para llevar un control de los préstamos y revisar si un libro se puede prestar o no.

Por otra parte, también mantiene un registro de sus empleados en todas las áreas de la biblioteca, cada empleado cuenta con Nombre, Edad, Puesto que desempeña y una ID de empleado. Un empleado tiene la oportunidad de sacar libros de la biblioteca por lo que se utiliza su ID al momento de hacer el préstamo.

Registro

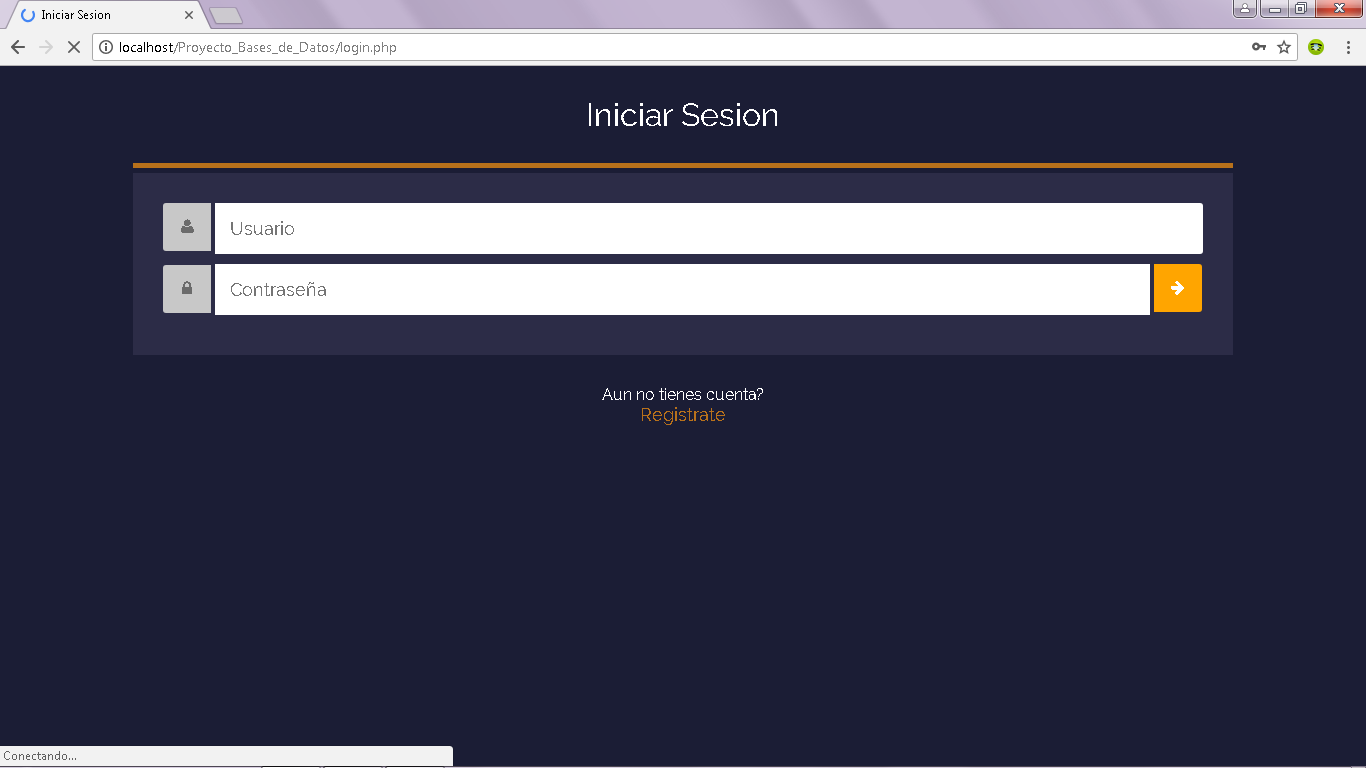


Para el registro se pedirá al usuario su nombre completo, nombre de usuario, correo contraseña.

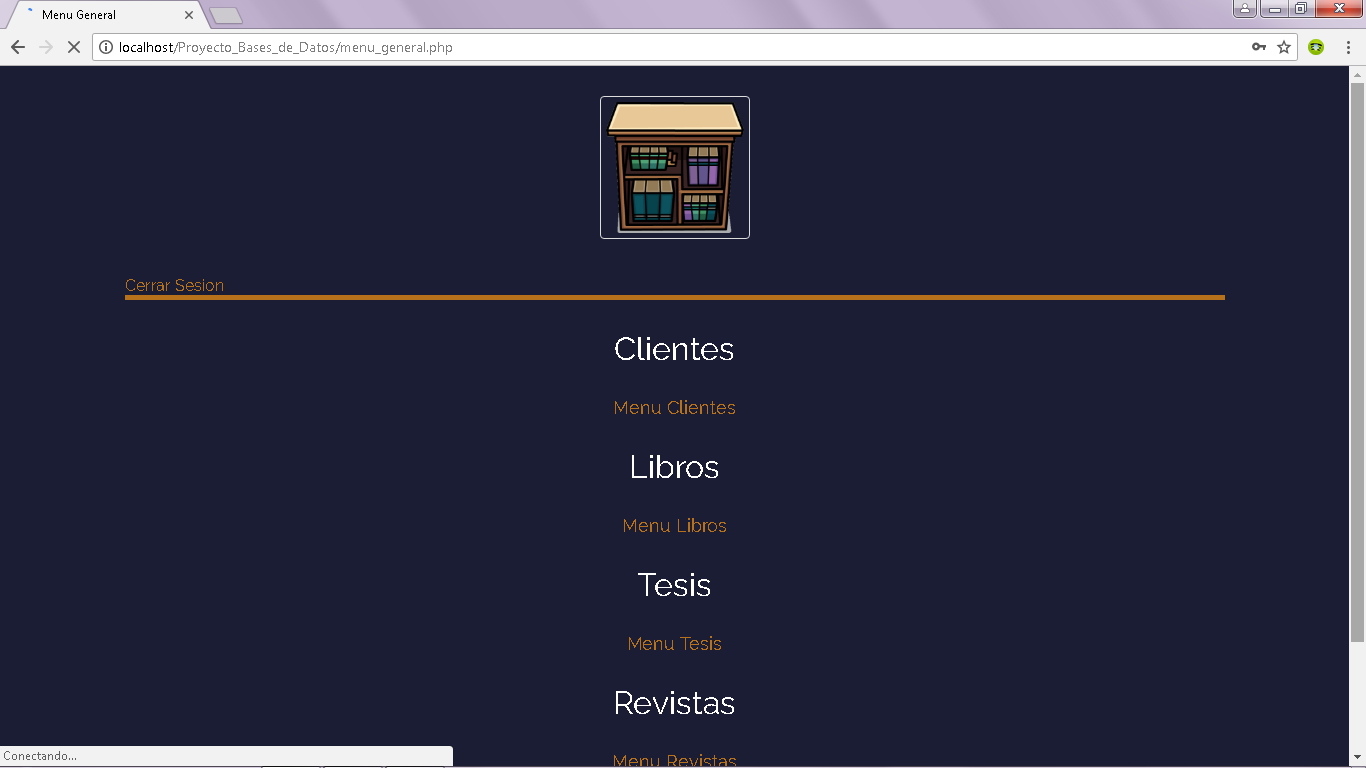
También se pedirá que el usuario ingrese la posición que cumple en la escuela (estudiante, egresado, maestro, administrativo).

Si el nombre de usuario ya existe, se pedirá volver a ingresar los datos.

Si el usuario ya tiene una cuenta creada dar click en iniciar sesión.



Menú general

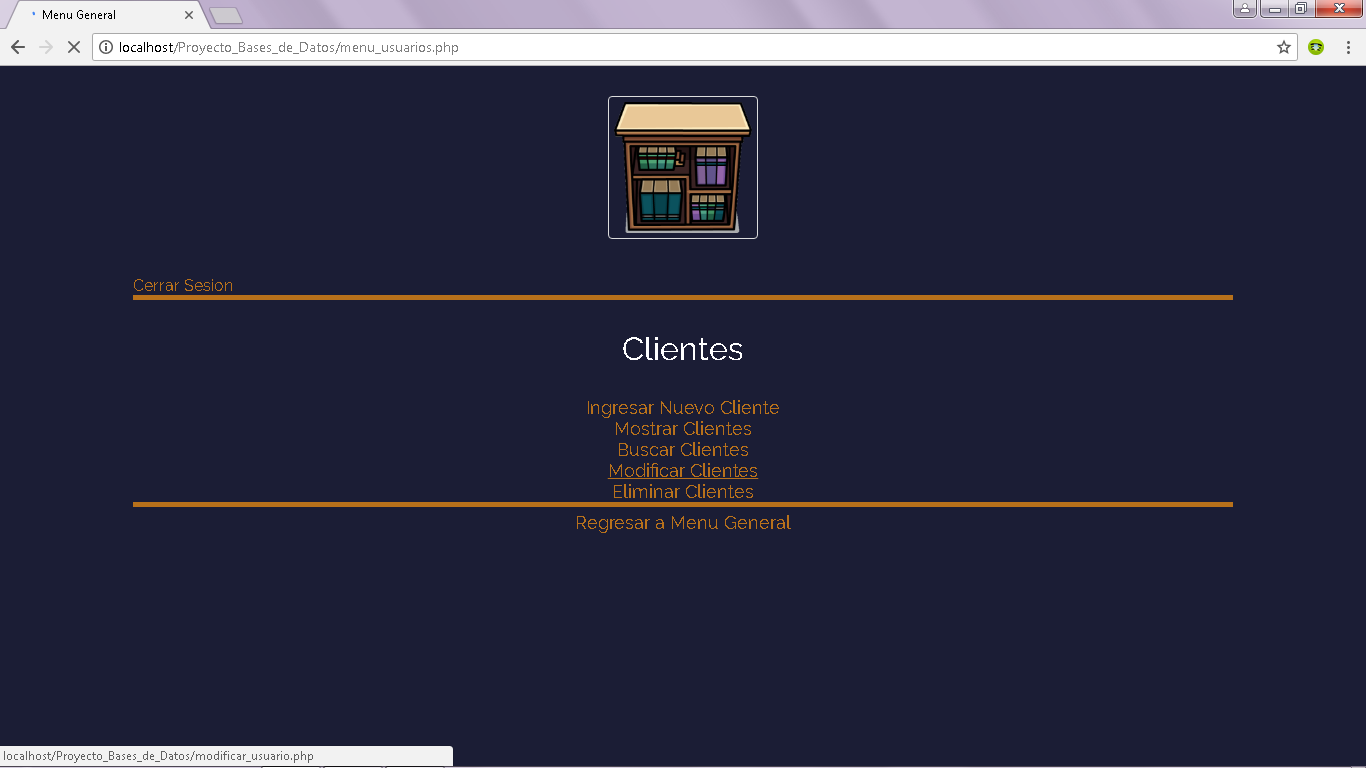


En el menú general se muestran diferentes menús a elegir (clientes, libros, tesis, revistas y prestamos) cada uno con sus operaciones correspondientes.

Al dar click en alguna de las opciones este te llevara al menú de esa opción

También desde el menú general podemos cerrar la sesión actual.

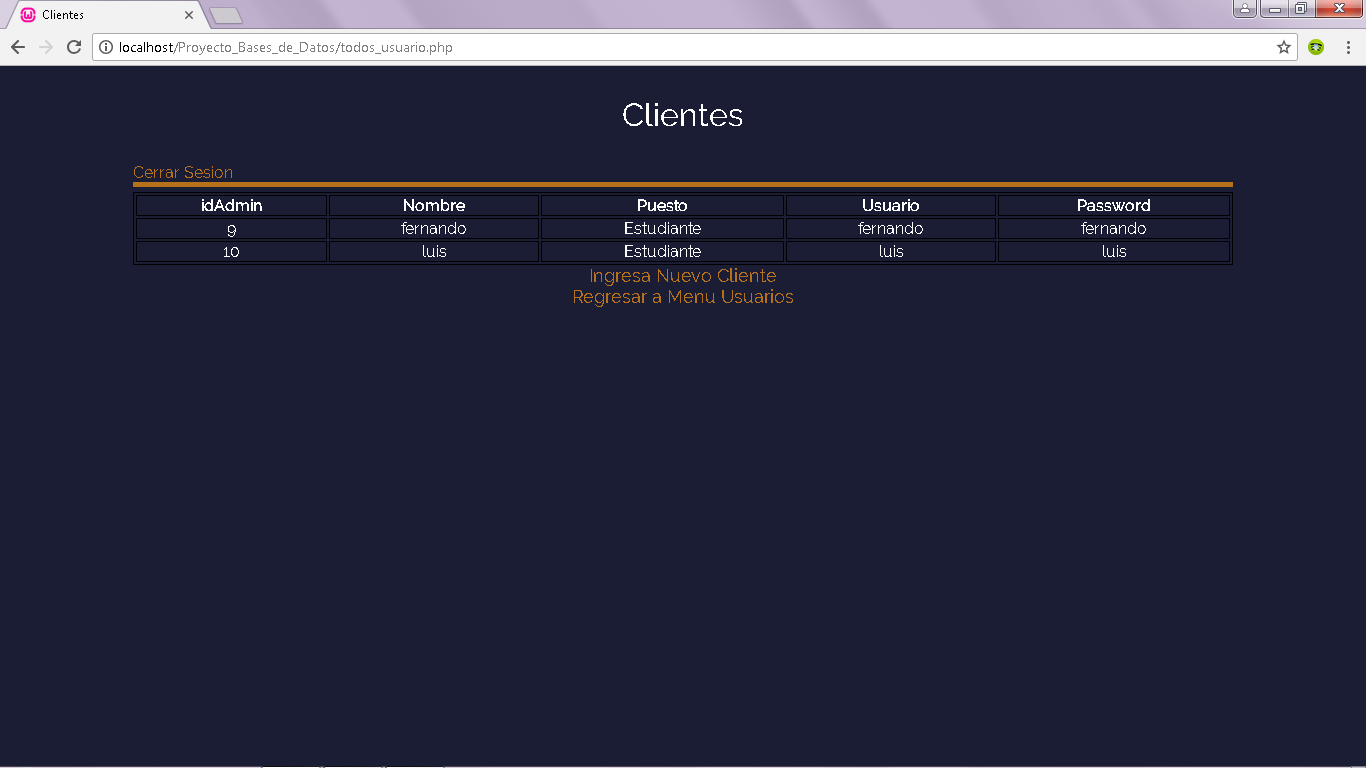
Menú clientes



En este menú se puede ingresar, mostrar, eliminar, modificar y mostrar clientes, asi como un link para regresar al menú general.

Mostrar

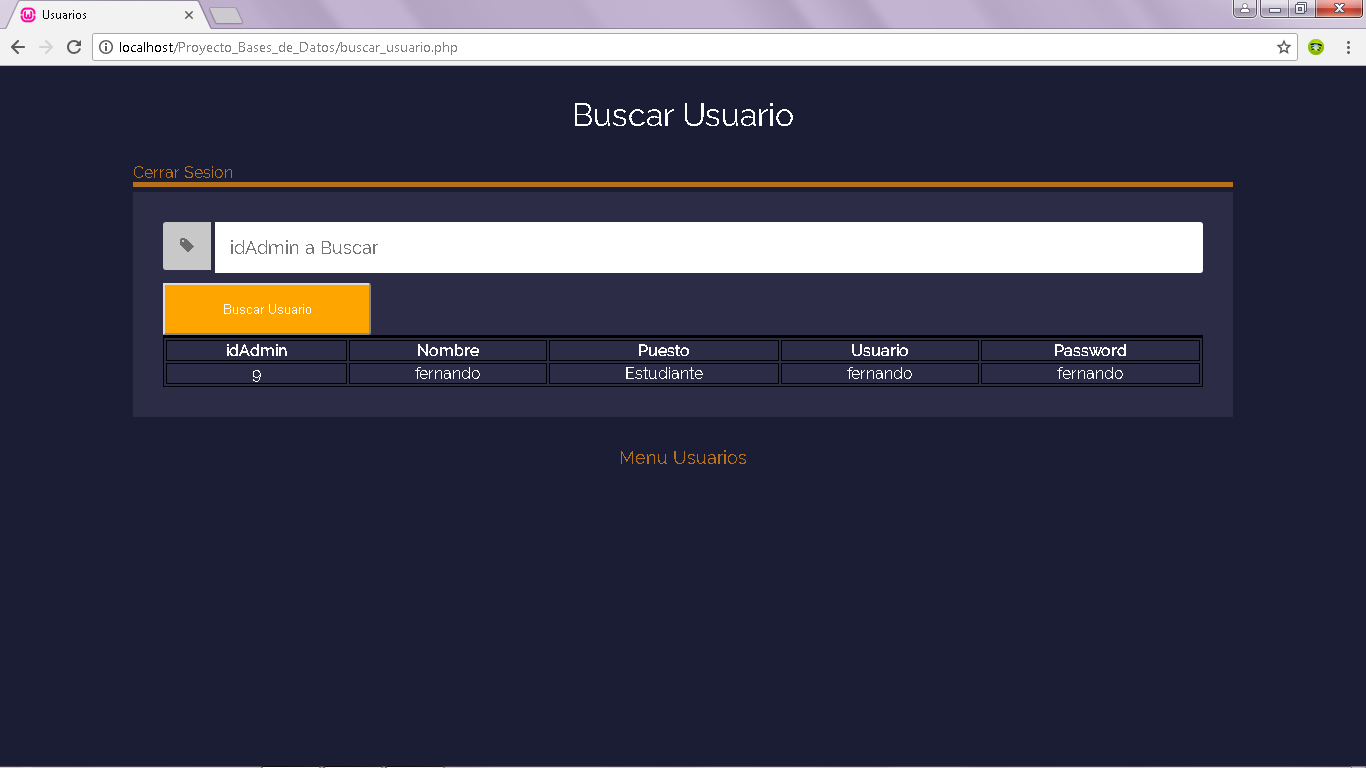
Al dar click en mostrar clientes se mostrarán automáticamente todos los usuarios registrados en la base de datos



Buscar usuario

Al buscar un usuario se debe ingresar el ID, si el Id no se encuentra aparecerá un mensaje indicando.

Si el ID coincide se mostrará el usuario correspondiente

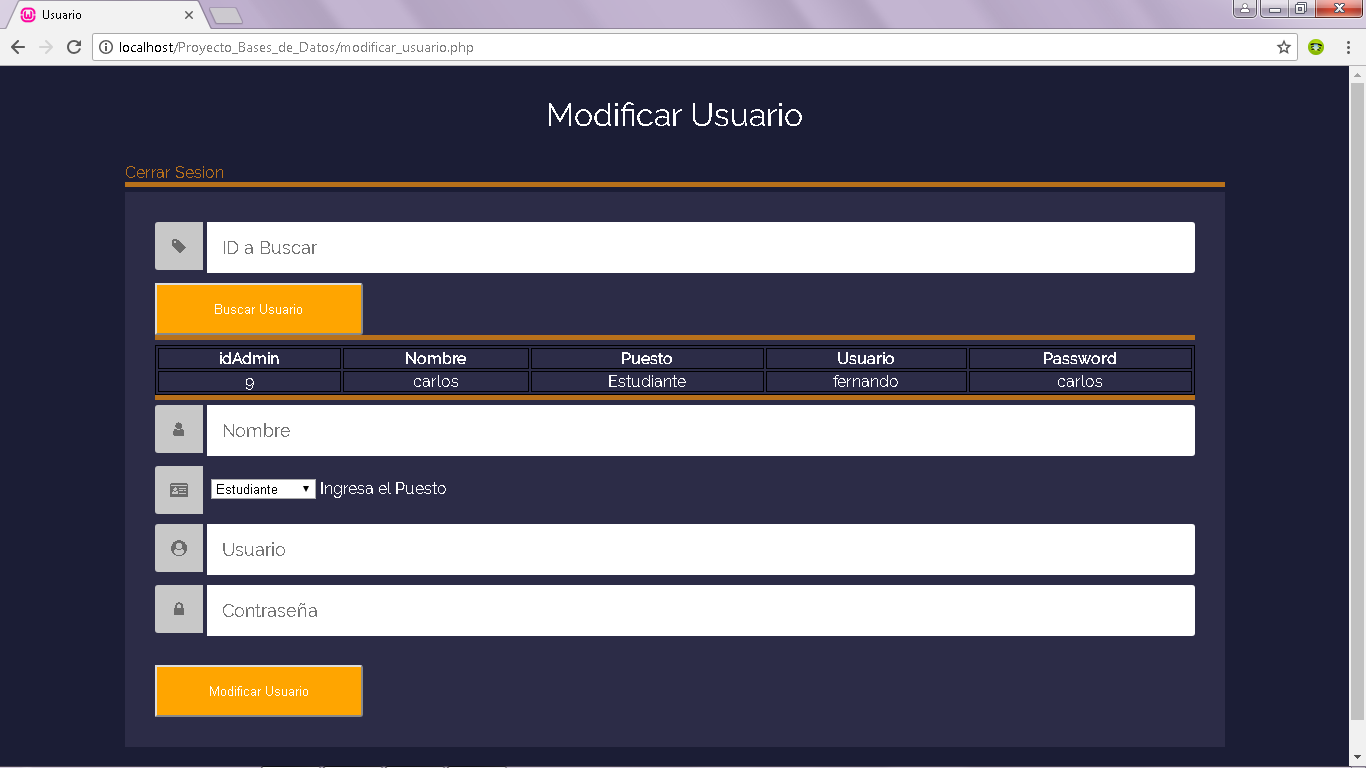


Modificar

Al modificar un usuario se debe ingresar el ID, si el Id no se encuentra aparecerá un mensaje indicando.

Si el ID coincide se mostrará el usuario correspondiente.

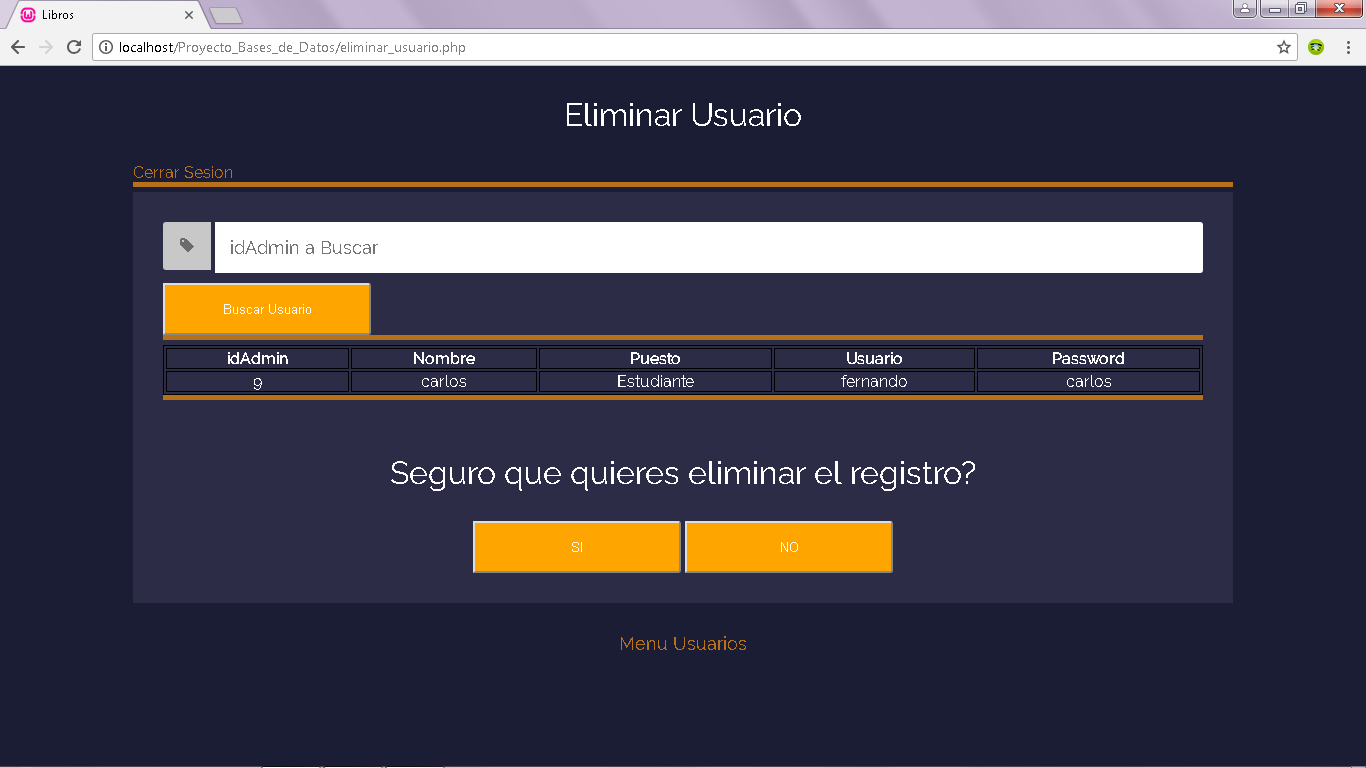
También se mostrará en la interfaz los campos para ingresar los nuevos datos del usuario, para confirmar los datos modificados hacer click en modificar usuario.



Eliminar

Al eliminar un usuario se debe ingresar el ID, si el Id no se encuentra aparecerá un mensaje indicando.

Si el ID coincide se mostrará el usuario correspondiente, asi como un mensaje de altera pidiendo al usuario la confirmación para eliminar al usuario.



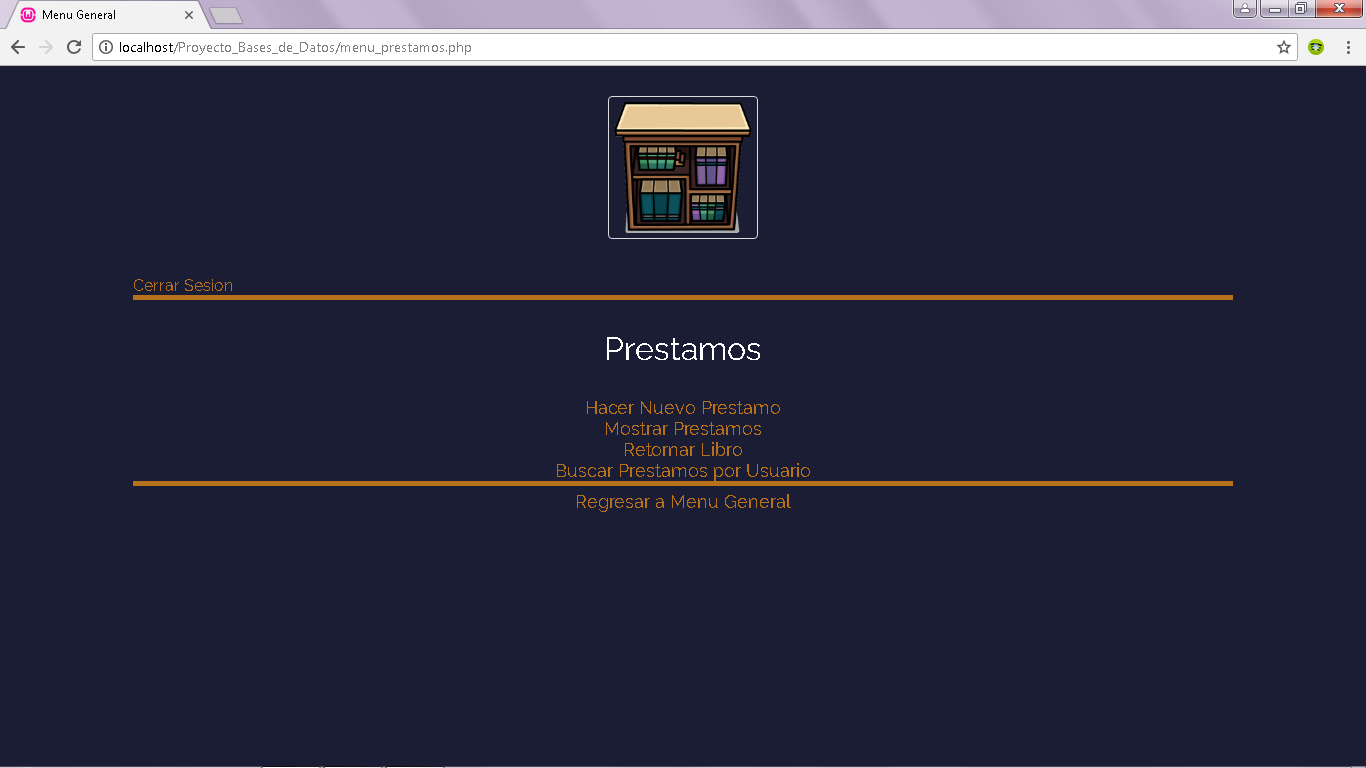
General

Para los menús de tesis, revistas y libros las operación de ingresar, modificar, eliminar, mostrar y buscar funcionan de la misma manera que la de usuarios (cada menú pidiendo sus datos específicos) .

Prestamos

el menú préstamo contiene las opciones:

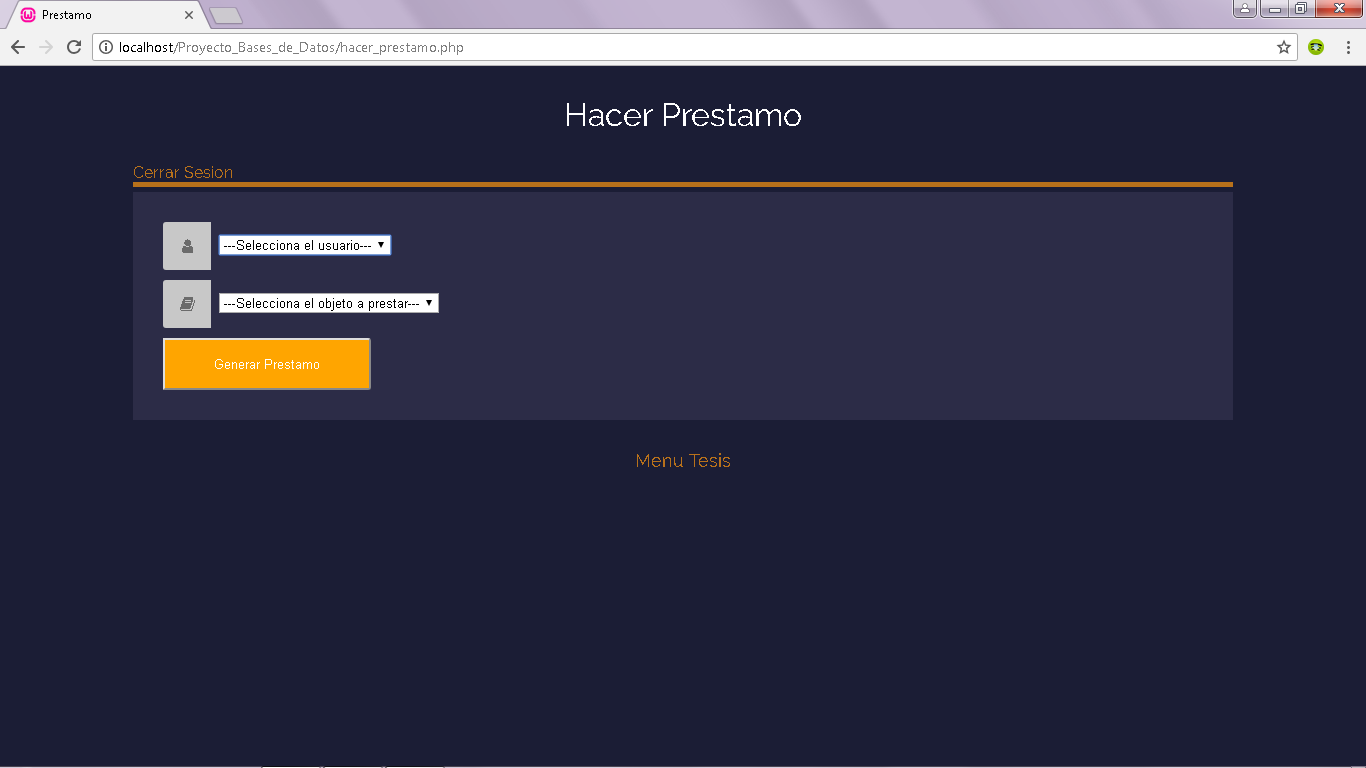
hacer préstamo , mostrar prestamos, retornar y buscar prestamos.



Hacer préstamo

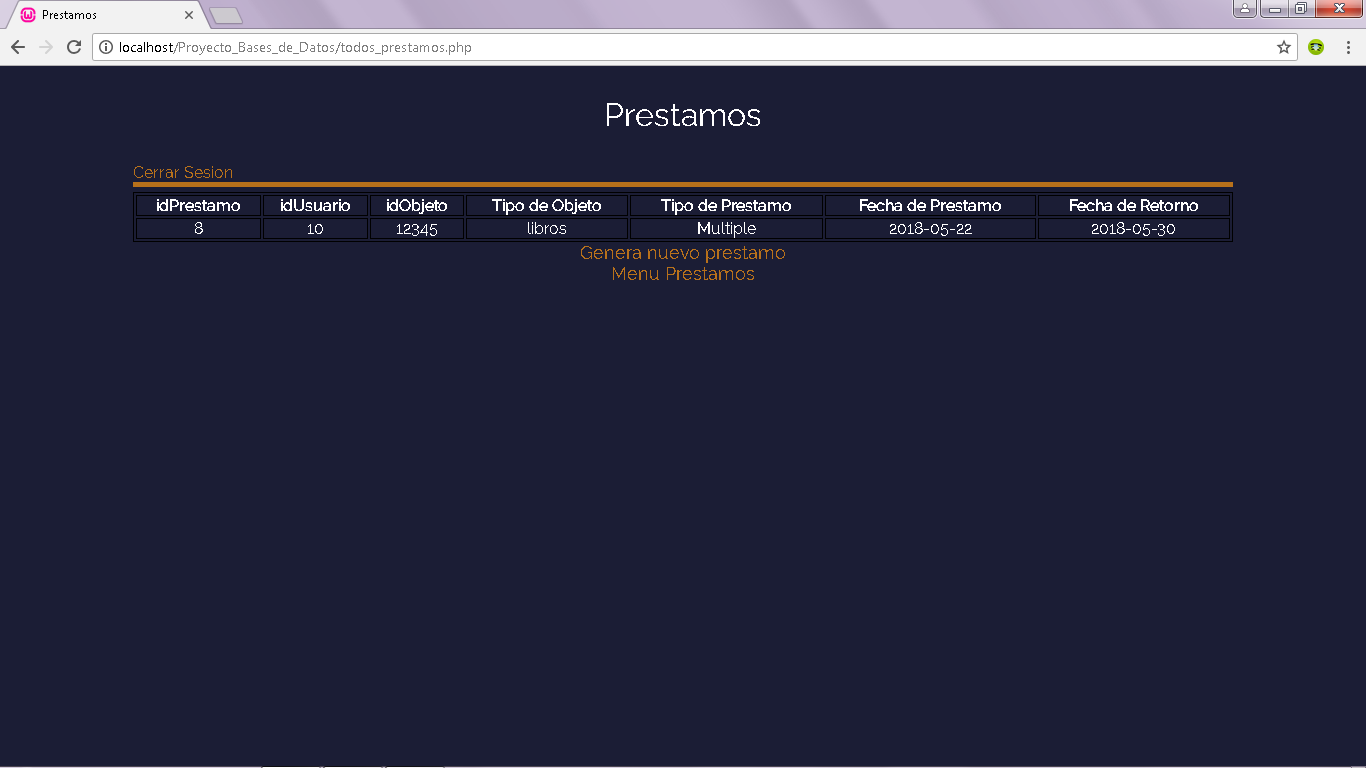
Al hacer un préstamo se debe seleccionar el usuario al que se le prestara el documento, así como seleccionar el documento deseado.

Confirmar para generar el préstamo.



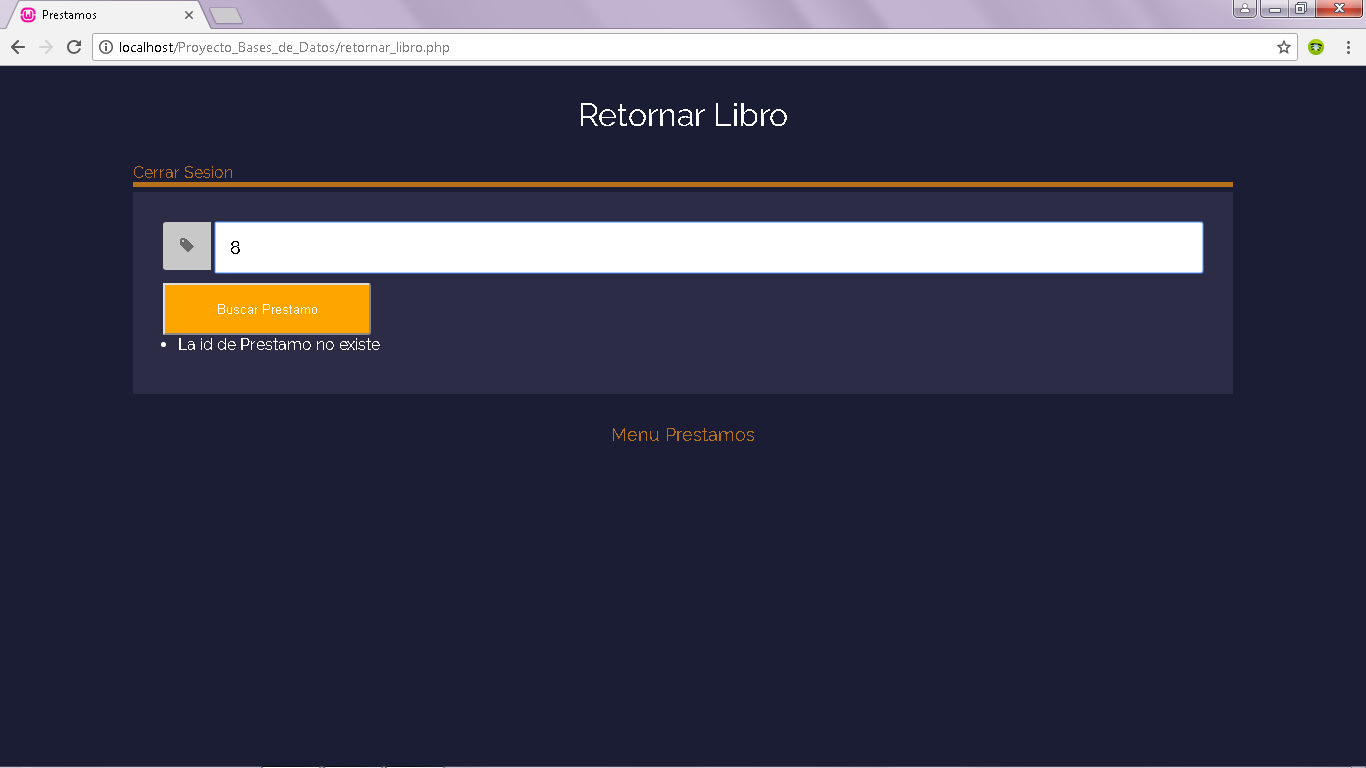
Mostrar prestamos

Al dar click en mostrar prestamos se muestran automáticamente todos los prestamos hechos en la biblioteca .



Retornar

Para regresar el documento prestado se debe ingresar el ID del préstamo efectuado.

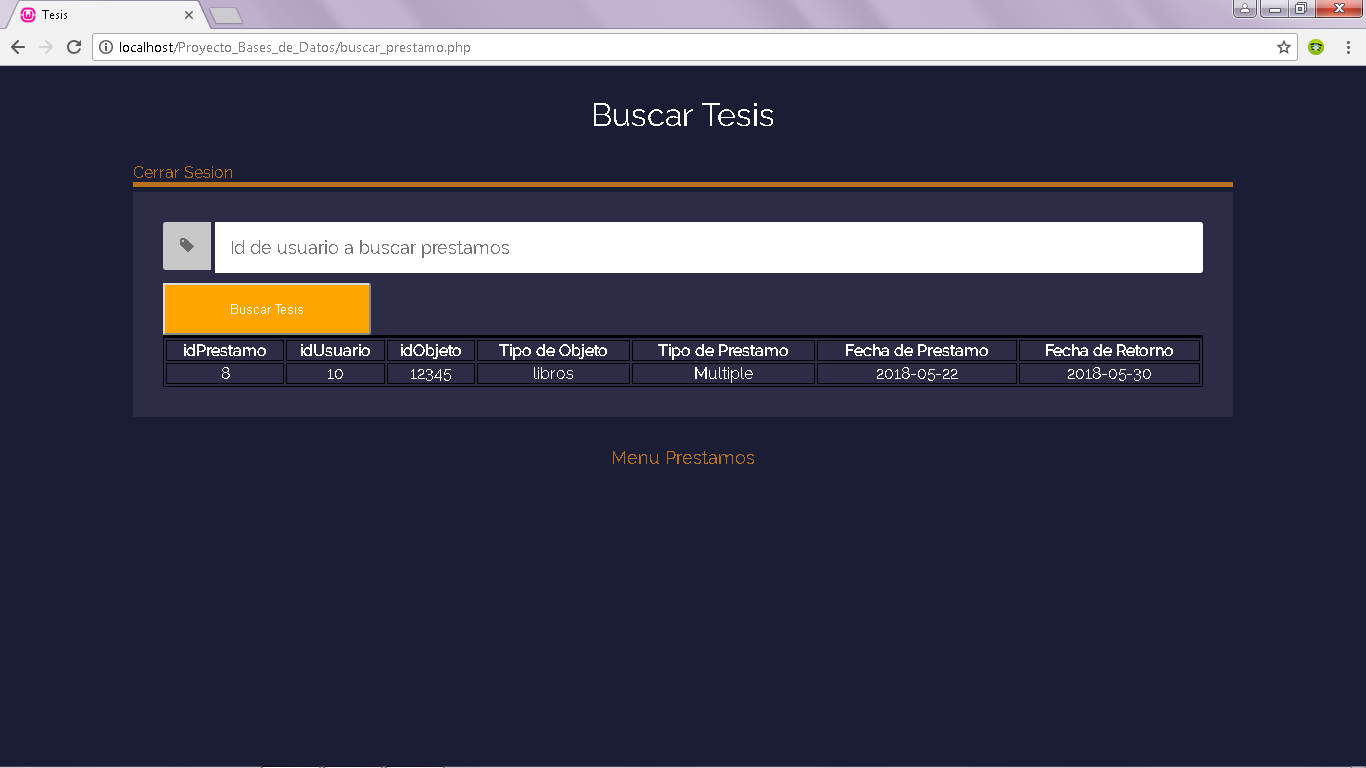


Si lo encuentra mostrara los datos del préstamo y la confirmación de que paga (si existe)la penalización.



Buscar prestamos por usuario

Para buscarlos préstamos que un usuario tiene se ingresa el ID del usuario, si se encuentra se mostraran los prestamos que el usuario ha realizado.



Así termina el manual del usuario