

**Índice**

Diagrama entidad relación…………………………………………………………………...4

[Introducción 3](#_heading=h.30j0zll)

[Requerimientos del Cliente 3](#_heading=h.1fob9te)

[Análisis 4](#_heading=h.3znysh7)

[Diagrama de Entidad-Relación 4](#_heading=h.2et92p0)

[Mapeo a Base de Datos 4](#_heading=h.tyjcwt)

[Casos de Uso 4](#_heading=h.3dy6vkm)

[Modelo de Casos de Uso 5](#_heading=h.1t3h5sf)

[Diseño 5](#_heading=h.4d34og8)

[Estándares del sistema 5](#_heading=h.2s8eyo1)

[Usabilidad 6](#_heading=h.17dp8vu)

[Eficiencia 6](#_heading=h.26in1rg)

[Robustez 6](#_heading=h.lnxbz9)

[Flexibilidad 6](#_heading=h.35nkun2)

[Seguridad 6](#_heading=h.1ksv4uv)

[Prototipos de Pantallas 6](#_heading=h.44sinio)

[Características 8](#_heading=h.2jxsxqh)

[Implementación 9](#_heading=h.z337ya)

[Resumen funcionalidad Loguin 9](#_heading=h.3j2qqm3)

[Resumen funcionalidad 2 (Reemplazar “2” por el nombre de la funcionalidad, ej: Gestión Cuotas) 10](#_heading=h.3whwml4)

[Resumen funcionalidad X (Un resumen por cada funcionalidad que abarque el sistema - Reemplazar “X” por el nombre de la funcionalidad, ej: Gestión Gastos) 10](#_heading=h.1y810tw)

[Cierre 10](#_heading=h.4i7ojhp)

[Retrospectiva del Proyecto 10](#_heading=h.2xcytpi)

[Anexo: Calificaciones 11](#_heading=h.1ci93xb)

Acomodar el índice y darle otra estética a la carátula

# Introducción

El cliente necesita que el sistema organice los muebles de la escuela

Nuestro cliente es la escuela proa, es un instituto escolar público con énfasis en el desarrollo de software

Nuestro programa se llama inventario mobiliario proa (I.M.P). su función es encargarse de los elementos que se encuentran en la institución proa, y poder llevar un registro de los elementos disponibles

Para construir este sistema usaremos

Editor de código: visual studio code

Lenguaje de programación: JavaScript

Motor BD: MySQL

Objetivo: solucionar el problema de la institución y poder llevar un registro de los elementos

# Requerimientos del Cliente

* Registrar un elemento nuevo (que el cliente pueda agregar un objeto que todavía no existe en el sistema con las características de marca, modelo, número de serie, lote, cantidad, nominación y fecha)
* Consultar elemento (que el cliente pueda acceder fácilmente a los datos y características del elemento)
* Modificar un elemento
* Eliminar un elemento
* Datos: características: marca, número de serie, lotes, cantidad, nominación

# Análisis

## Diagrama de Entidad-Relación



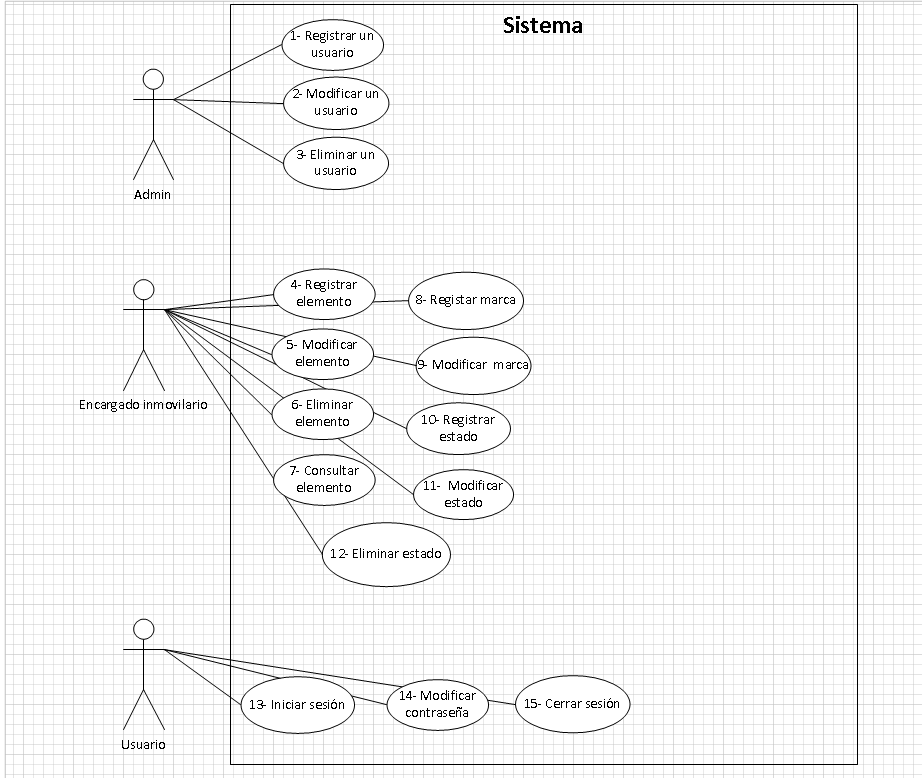
## Mapeo a Base de Datos



## Casos de Uso

1. Registrar usuario
2. Modificar usuario
3. Eliminar usuario
4. Registrar elemento
5. Modificar elemento
6. Eliminar elemento
7. Consultar elemento
8. Registrar marca
9. Modificar marca
10. Registrar estado
11. Modificar estado
12. Eliminar estado
13. Iniciar sesión
14. Modificar contraseña
15. Cerrar sesión

## Modelo de Casos de Uso



# Diseño

## Estándares del sistema

### Usabilidad

Este sistema contará con una buena capacidad de adecuación, al usuario se le hará sencillo el utilizar el programa, podrá ser utilizado con facilidad, podrá eliminar o modificar elementos si los pone de forma errónea, será de linda estética, no estará pensado para determinadas características y discapacidades.

### Eficiencia

Nuestro proyecto cumple con su objetivo con una gran eficacia y precisión a la hora de agregar elementos o eliminar de su stock.

El tiempo estimado para que una persona poco experimentada pueda cumplir su objetivo dentro del programa es de 2:30 m en cambio una persona ya experimentada dentro del programa se estima que en un tiempo de 1:20 puede llegar a cumplir su objetivo.

### Robustez

Será un sistema que no producirá errores al modificarlo, se podría modificar con sencillez.

### Flexibilidad

Nuestro programa se va a adaptar fácilmente a los cambios propuestos por los usuarios para poder cumplir con sus necesidades.

### Seguridad

En nuestro sistema se podrá comprobar la identidad del usuario con un nombre de usuario y su respectiva contraseña, su nombre y apellido también serán necesarios.

Cada usuario solamente tendrá acceso a la información que le corresponda, tendrá que acceder solo a las pantallas correspondiente a su rol,

### LOS ESTÁNDARES PROPIOS DEL SISTEMA

TIPO DE LETRA: ROBOTO CONDENSED, sans-serif de Google Font

PALETA DE COLORES

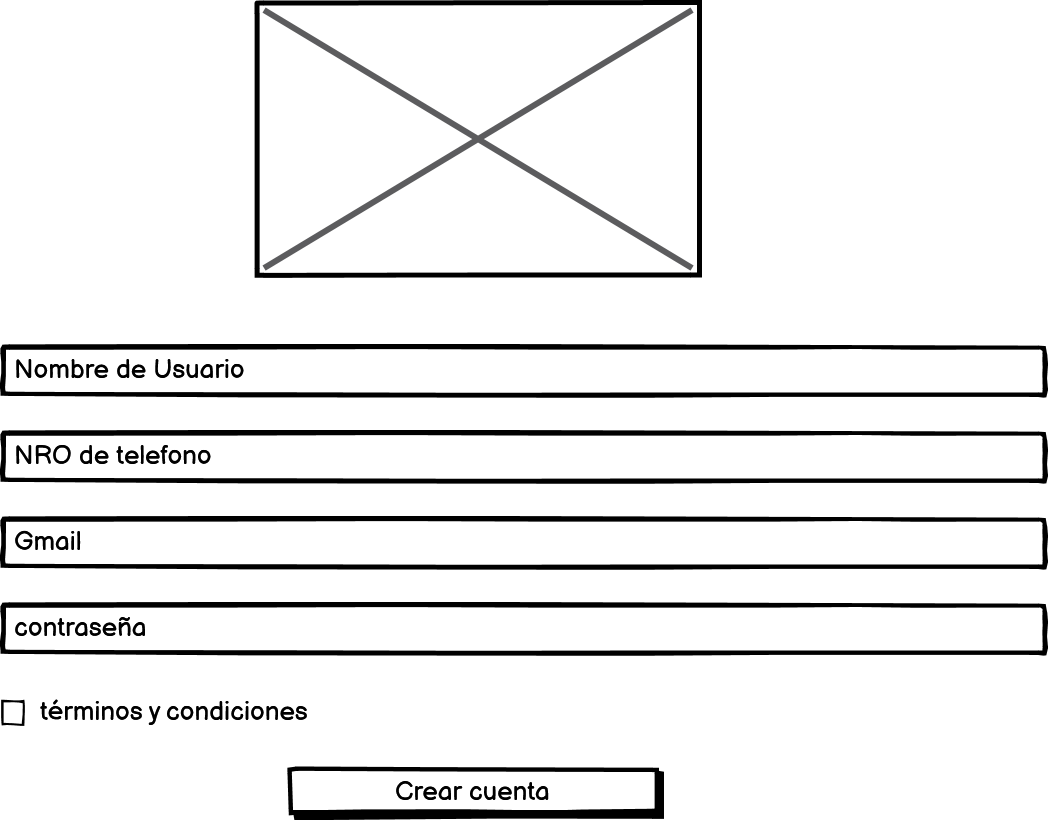
Patrón de fondo

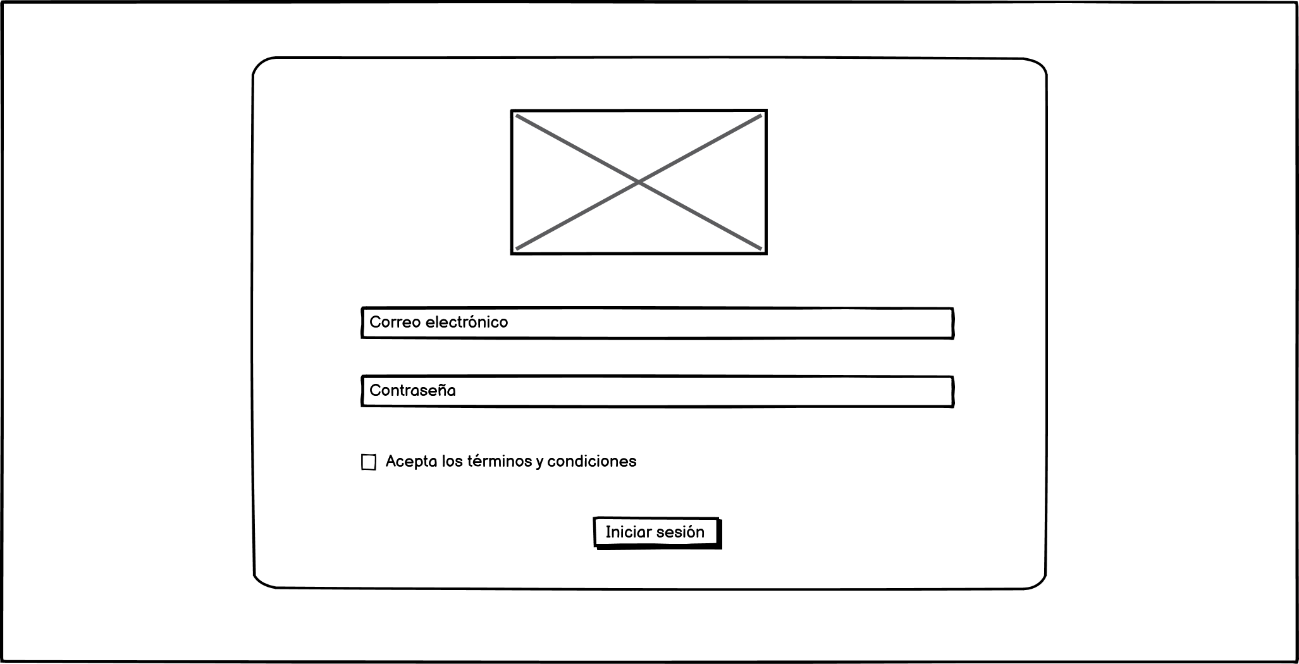
Descripción generada automáticamente

#### TAMAÑO DE COMPONENTES: La Mayoría del alto de los componentes es de 24 px Y el ancho en general es de 300 px

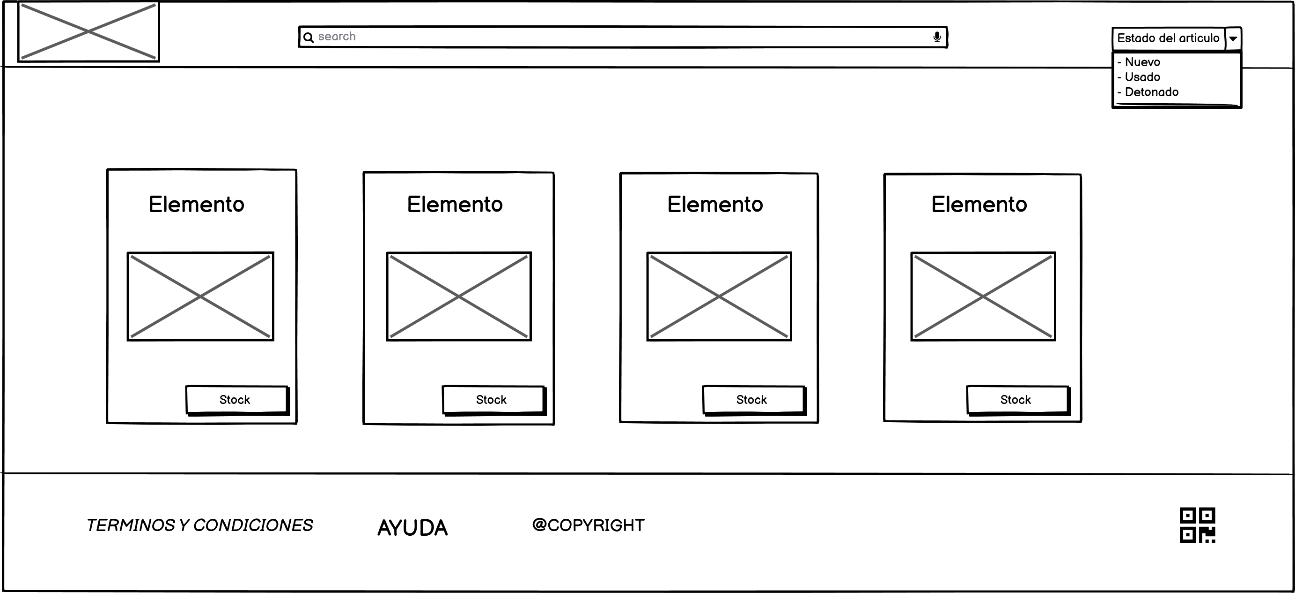
## Prototipos de Pantallas

Registro

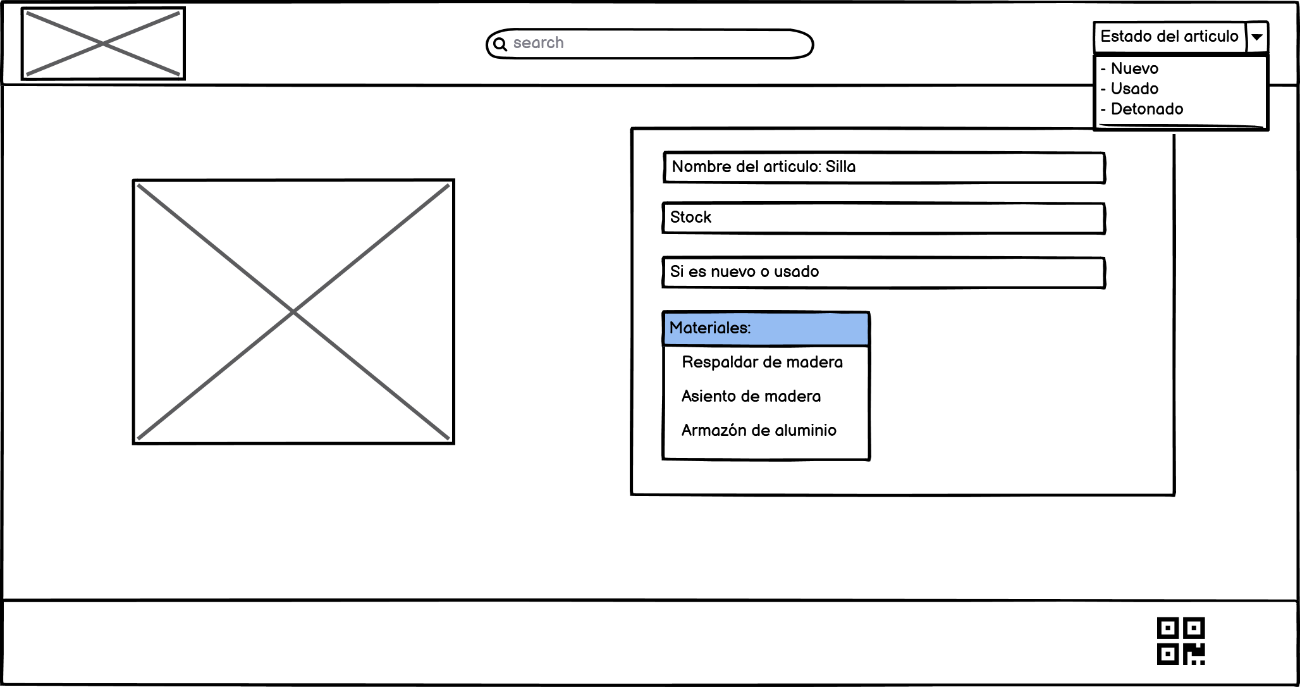


Iniciar sesión

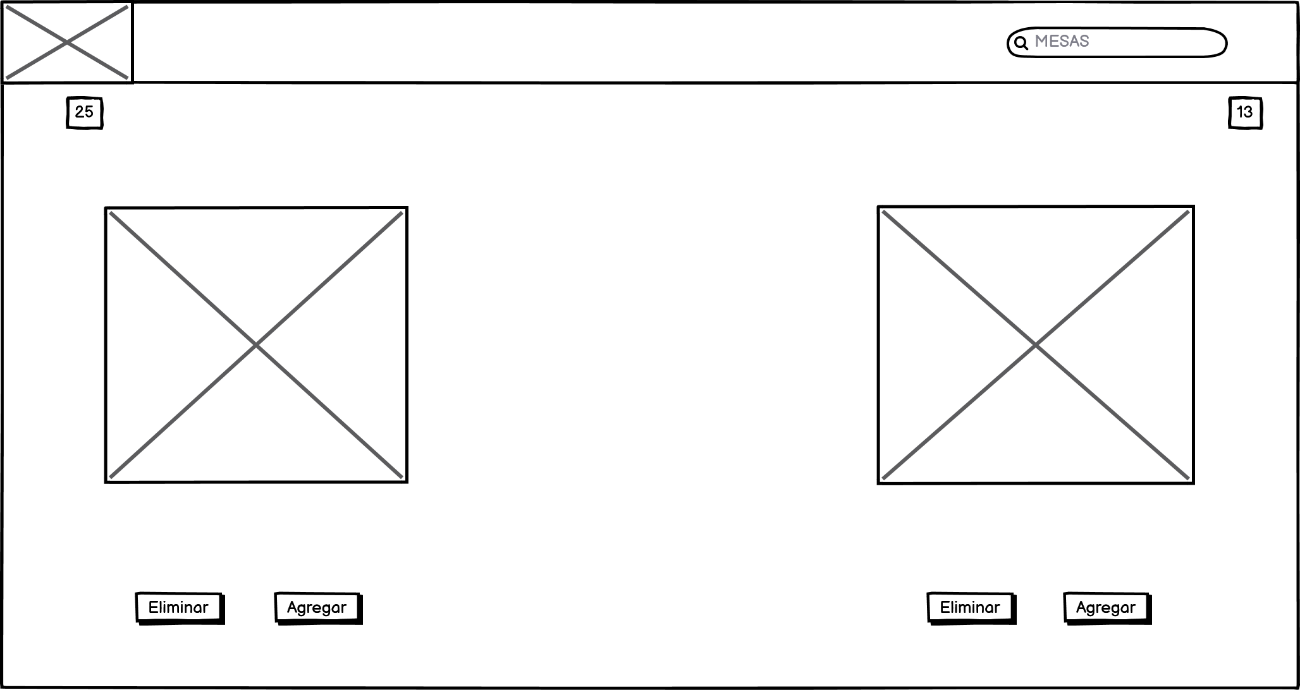
Main



# Características



Agregar o eliminar



Diagrama

Descripción generada automáticamente

# Implementación

## Resumen funcionalidad Login

Esta función logra crear una interfaz gráfica en HTML, CSS y JavaScript. También se conecta a la base de datos MySQL con node.js.

La información la consigue mediante unos inputs donde el usuario ingresa esos mismos datos allí cuando hace clic en el botón enviar los datos se envían a la base de datos en la tabla usuarios.

Luego para inicia sesión se compara los datos que ingrese el usuario en los campo de email y contraseña. Estos mismos datos se comparan con los que tiene la base de datos si coinciden ingresa si no, no ingresa

-Lo que se propuso a nivel sistema pasa esta funcionalidad si se logro

-Nuestro trabajo conto con un buen trabajo en equipo, pero estamos tratando de mejora es ese aspecto

-Si , estamos conforme con nuestro trabajo en grupo y los dichos resultados

-Los aspectos del trabajo en equipo que debemos mantener es la división de las tareas de cada uno para poder progresa todos juntos

# Función de insertar, eliminar y editar elementos

En esta función logramos que todos los elementos que el usuario quería eliminar, ingresar o editar lo puede hacer en cuanto al trabajo en equipo en esta ocasión logramos trabajar mejor como equipo dividiéndolos tareas de forma más equitativa y gracias a este trabajo en equipo hicimos que el código funcione correctamente.

Podríamos mejorar en el aspecto de la comunicación del grupo, creemos que aportar más ideas estaría mejor para el avance del programa.

Con el tema de la creatividad creemos que esta básico el diseño, le haría falta más cosas como carteles bordes a los botones etc. Algo que le mejore la estética.

## Resumen funcionalidad protección de rutas

En esta funcionalidad se logró proteger la ruta cuando el usuario no está logueado y también este sitió fue adaptado a casi todo los estilos de pantalla, tanto como para computadora, tablet y celular.

Se realizó el despliegue de la aplicación en beta.

De acuerdo a la estética del programa, se le realizaron algunos cambios de colores layout.

# Cierre

## Retrospectiva del Proyecto

(Análisis de los logros alcanzados en el desarrollo del sistema completo, tanto a nivel sistema como a nivel equipo, qué aprendieron a lo largo del proyecto, en qué les gustaría mejorar, qué les hubiese gustado implementar que no pudieron, si cumplieron las metas propuestas, si cambiarían algo del proyecto, etc.)

# Anexo: Calificaciones

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Primer Entrega Trabajo Final Integrador | | | |  |  |  |  |
|  |  |  |  | Calificación por Estudiante | | | |
| Espacio Curricular | Aprendizaje | Aplicado | Observación | Galliano Donato | Romero Benjamín | Salomón Martín | Vivas Florencia |
| Programación II, Diseño de Interfaces y Usabilidad | Reconocimiento de que las elecciones que hacen las personas cuando crean programas, aplicaciones y sistemas se basan en intereses personales, experiencias y necesidades. | Sí |  | 8 | 8 | 8 | 8 |
| Trabajo colaborativo con los compañeros. | Sí |  |
| Incorporación de estrategias para trabajar con compañeros que poseen diferentes habilidades y puntos de vista. | Sí |  |
| Presentación de una idea específica basándose en la información obtenida para la resolución del problema. | Sí | Con observaciones |
| Representación de los datos e información obtenida utilizando gráficos, diagramas o presentaciones. | Sí | Con observaciones |
| Utilización del lenguaje adecuado para referirse a conceptos y dispositivos, refiriéndose con la terminología correspondiente. | Sí |  |
| Presentación de manera clara de los objetivos y las productos esperados de las soluciones desarrolladas. | Sí |  |
| Definición de roles en el trabajo en equipo para, una vez identificadas las partes del problema, poder trabajar en la resolución de los subproblemas. | Sí | Con observaciones |
| Diseño de Interfaces y Usabilidad | Identificación de los componentes fundamentales en el diseño de interfaces gráficas modernas. | Sí |  | 8 | 8 | 8 | 8 |
| Comprensión e identificación de los diferentes pasos del proceso de desarrollo de interfaces gráficas de usuario. | Sí | Con observaciones |
| Reconocimiento de la importancia de la existencia de estándares nacionales e internacionales para el desarrollo y creación de interfaces. | Sí |  |
| Identificación, de manera general, de los diferentes roles que forman parte del proceso de desarrollo del software. | Sí |  |
| Comprensión del concepto de Usabilidad, Eficiencia y Robustez en cuanto al desarrollo de interfaces. | Sí | Con observaciones |
| Comprensión de la importancia de los Casos de Uso y sus componentes para el diseño de interfaces. | Sí |  |
| Definición de casos de uso básicos para el proceso de diseño de interfaces. | Sí | Con observaciones |
| Identificación de tipografías, colores necesarios para generar una interfaz adecuada al producto a desarrollar. | No |  |
| Identificación de principios básicos de la usabilidad (Facilidad de Aprendizaje-Facilidad de Uso-Flexibilidad: Robustez). | Sí | Con observaciones |
| Incorporación del concepto de seguridad informática. | Sí |  |
| Comprensión de la importancia de definir prototipos de interfaces, previo a comenzar la implementación de la interfaz. | Sí | Con observaciones |
| Creación de prototipos de interfaces, ante una situación problemática planteada, que vayan presentando posibles soluciones a las vistas de usuario. | Sí |  |
| Identificación y definición de casos de uso para el diseño de la interfaz. | Sí | Con observaciones |
| Identificación de si en los casos de usos definidos se tienen en cuenta todos los posibles usos que el usuario puede dar a la plataforma. | Sí |  |
| Respeto por las convenciones existentes para la definición de ciertos aspectos de las interfaces. | Sí |  |
| Estructuras y Almacenamiento de Datos III | Incorporación de la idea central de los Diagramas de Entidad Relación. | Sí |  | 8 | 8 | 8 | 8 |
| Desarrollo de noción de funcionalidad e importancia del uso de Diagramas de Entidad - Relación. | Sí |  |
| Comprensión del concepto y función de Entidad. | Sí |  |
| Comprensión del concepto y función de los atributos. | Sí |  |
| Identificación, comprensión y establecimiento de los atributos de una entidad en particular. | Sí |  |
| Representación de ciertos objetos simples de la vida cotidiana como entidades, utilizando los atributos necesarios para generar las entidades deseadas. | Sí |  |
| Comprensión de que es una clave primaria y su función dentro de los diagramas de Entidad - Relación. | Sí |  |
| Comprensión de que es una clave foránea y su función dentro de los diagramas de Entidad - Relación. | Sí |  |
| Diferenciación de clave primaria y clave foránea. Identificación de cuándo es necesario utilizar cada una de estas claves. | Sí |  |
| Definición de claves primarias que sean capaces de identificar de manera exclusiva cada objeto perteneciente a una entidad. | Sí | Con observaciones |
| Identificación, comprensión y establecimiento de las claves primarias asociadas a entidades de nuestro diagrama Entidad -Relación. | Sí | Con observaciones |
| Aplicación de un criterio apropiado para la elección de claves foráneas en diagramas Entidad - Relación. | Sí |  |
| Comprensión del concepto de Relación, y su función dentro de los diagramas de Entidad - Relación. | Sí |  |
| Definición de relaciones entre entidades. | Sí |  |
| Diferenciación de tipos de relaciones (1-1, 1-\*, \*-1, \*-\*). | Sí |  |
| Identificación en situaciones problemáticas planteadas de las entidades que conformen las mismas y el tipo de relaciones entre las entidades | Sí |  |
| Comprensión de los diagramas de entidad - relación como herramientas que nos permiten representar de forma sencilla situaciones y elementos reales, para que puedan ser interpretados por una computadora. | Sí |  |
| Representación de objetos utilizando entidades y sus atributos. | Sí | Con observaciones |
| Utilización de nombres entidades relacionados con el objeto que representen | Sí |  |
| Elección de nombres no extensos y relacionados al rol de los atributos. | Sí |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Segunda Entrega Trabajo Final Integrador | | | |  |  |  |  |
|  |  |  |  | Calificación por Estudiante | | | |
| Espacio Curricular | Aprendizaje | Aplicado | Observación | Galliano Donato | Romero Benjamín | Salomón Martín | Vivas  Florencia |
| Programación II | Reconocimiento de que las elecciones que hacen las personas cuando crean programas, aplicaciones y sistemas se basan en intereses personales, experiencias y necesidades. | Sí |  | B | B | B | B |
| Trabajo colaborativo con los compañeros. | Sí | Escaso | R- | R- | R- | R- |
| Incorporación de estrategias para trabajar con compañeros que poseen diferentes habilidades y puntos de vista. | Sí | Escaso | R- | R- | R- | R- |
| Presentación de una idea específica basándose en la información obtenida para la resolución del problema. | Sí |  | R | R | R | R |
| Representación de los datos e información obtenida utilizando gráficos, diagramas o presentaciones. | Sí |  | B | B | B | B |
| Utilización del lenguaje adecuado para referirse a conceptos y dispositivos, refiriéndose con la terminología correspondiente. | Sí |  | B- | B- | B- | B- |
| Presentación de manera clara de los objetivos y las productos esperados de las soluciones desarrolladas. | Sí |  | B- | B- | B- | B- |
| Definición de roles en el trabajo en equipo para, una vez identificadas las partes del problema, poder trabajar en la resolución de los subproblemas. | No |  | \_ | \_ | \_ | \_ |
| Comprensión del concepto y funcionalidad de Clase. | No |  | \_ | \_ | \_ | \_ |
| Comprensión del concepto de Atributo e identificación de su funcionalidad. | No |  | \_ | \_ | \_ | \_ |
| Definición de clases simples que representen objetos cotidianos, definiendo atributos específicos que permitan caracterizar de manera correcta a estos elementos. | Sí |  | R- | R- | B | R- |
| Uso de nombres que se relaciones con la función de la clase. Respeto por las normas correspondientes al lenguaje elegido para escoger el nombre de las clases. | Sí |  | R- | R- | B | R- |
| Uso de nombres y tipos de datos adecuados a su contenido al definir los atributos de las clases. | Sí |  | R | R | B | R |
| Creación del método de inicialización/constructor de clase, teniendo en cuenta qué atributos deben ser inicializados al momento de crear un objeto. | Sí |  | R | R | B | R |
| Reconocimiento de cuándo es necesario crear atributos o métodos privados o públicos. | No |  | \_ | \_ | \_ | \_ |
| Creación de métodos getters y setters. | No |  | \_ | \_ | \_ | \_ |
| Creación de métodos para definir el comportamiento de los objetos que se crearán a partir de la clase definida. | No |  | \_ | \_ | \_ | \_ |
| Identificación de errores de sintaxis en diferentes programas, según el lenguaje utilizado. | Sí |  | R | R | B | R |
| Identificación, ante la ejecución de un programa secuencial sencillo, de si funciona o no correctamente, teniendo en cuenta el objetivo del programa creado. | Sí |  | B | B | B | B |
| Utilización de nombres de clases, atributos y métodos relacionados a su objetivo. | Sí |  | R | R | B | R |
| Respeto por las convenciones definidas por los lenguajes de programación para definir nombre de clases, atributos y métodos. | Sí |  | R | R | B | R |
| Comprensión de que los programas creados son ejecutados por una computadora pero los desarrollamos para personas, por lo cual tienen que ser legibles para sus compañeros. | Sí |  | B | B | B | B |
| Comentario de las clases, atributos y métodos teniendo en cuenta sus funciones. | Sí | Escaso | R | R | R | R |
| Utilización de estrategias para generar feedback que aporte para la comprensión ante la pregunta de un compañero/a. | No |  | \_ | \_ | \_ | \_ |
| Reconocimiento de los beneficios de utilizar sistemas de control de versiones (git, svn, etc.) para el desarrollo de programas en equipo. | Sí |  | M | B | M | B |
| Uso de mensajes relacionados a la tarea resuelta cada vez que se realice un commit. | Sí | Escaso | R | R | B | R |
| NOTA | | | 7 | 7 | 7 | 7 |
| Diseño de Interfaces y Usabilidad | Identificación de los componentes fundamentales en el diseño de interfaces gráficas modernas. | Sí |  | B | R | M | B |
| Comprensión e identificación de los diferentes pasos del proceso de desarrollo de interfaces gráficas de usuario. | Sí |  | B | R | M | B |
| Identificación, de manera general, de los diferentes roles que forman parte del proceso de desarrollo del software. | No |  | \_ | \_ | \_ | \_ |
| Creación de interfaces utilizando plataformas actuales para su desarrollo. | Sí |  | B | R | M | B |
| Creación de interfaces utilizando conceptos de interfaz gráfica (arquitectura de la información, uso de los colores, tipografías, concepto de diseño). | Sí |  | B | R | M | B |
| Utilización de bibliotecas, frameworks desarrollados por terceros para poder desarrollar las interfaces diseñadas. | Sí |  | B | R | M | B |
| Utilización y comprensión del funcionamiento de funciones específicas para la implementación de interfaces. | Sí |  | B | R | M | B |
| Testeo de la interfaz desarrollada de forma exhaustiva para comprobar si funciona de la manera deseada. | Sí |  | B | R | B | B |
| Verificación de cuestiones de contenido dentro de la interfaz (ortografía, autenticidad del contenido, usabilidad, contrastes de colores, tamaños y tipos de fuentes) | Sí |  | B | R | B | B |
| Respeto por las convenciones existentes para la definición de ciertos aspectos de las interfaces. | Sí | Con observaciones | B | R | B | B |
| NOTA | | | 8 | 8 | 9 | 8 |
| Estructuras y Almacenamiento de Datos III | Traducción de entidades en tablas respetando las características de las entidades definidas. | Sí | Algunas | B | R | B | B |
| Agregado de forma coherente de diferentes elementos a tablas definidas. | No |  | \_ | \_ | \_ | \_ |
| Representación de diferentes relaciones en Tablas. | Sí |  | B | R | B | B |
| Incorporación de sintaxis y características de los lenguajes SQL. | No |  | \_ | \_ | \_ | \_ |
| Incorporación de instrucciones específicas para realizar consultas a la base de datos. | No |  | \_ | \_ | \_ | \_ |
| Realización de consultas simples a la base de datos. | No |  | \_ | \_ | \_ | \_ |
| Uso de Sentencias de Creación. | No |  | \_ | \_ | \_ | \_ |
| Uso de Sentencias de Lectura. | No |  | \_ | \_ | \_ | \_ |
| Uso de Sentencias de Actualización. | No | Todavía no se aplica |  |  |  |  |
| Uso de Sentencias de Eliminación. | No | Todavía no se aplica |  |  |  |  |
| Realización de Consultas Simples. | No |  | \_ | \_ | \_ | \_ |
| Incorporación de habilidades para realizar Consultas Combinadas. | No |  | \_ | \_ | \_ | \_ |
| Creación de una Base de Datos simple. | Sí |  | B | R | M | B |
| Creación de las tablas correspondientes a la base de datos creada | Sí | Algunas | B | R | B | B |
| NOTA | | | 3 | 3 | 3 | 3 |