



Proyecto 2

Escuela de Ciencias y Sistemas

Introducción a la programación
y computación 1

Ing. Neftali de Jesus Calderon

Mendez

Aux. Erwin Fernando Vásquez Peñate

OBJETIVOS

Objetivo general

Que el estudiante pueda aplicar los conocimientos del curso para crear una solución de software de un problema cercano a la realidad.

Objetivos específicos

- Que el estudiante pueda identificar una solución acorde a una necesidad.
- Que el estudiante pueda realizar una toma de requerimientos basado en una necesidad previamente identificada.
- Que el estudiante pueda implementar una solución a través de un lenguaje de programación.
- Entender la importancia de la interacción del software con el entorno real.

ENUNCIADO

Descripción general

El dojo de karate “Cobra Kai” ha crecido exponencialmente, a tal punto que el sensei del dojo ha decidido implementar una solución web para su dojo.

Problemas a resolver

De momento la aplicación con interfaz desarrollada en Java en la fase anterior ha funcionado correctamente, pero el sensei desea crear una solución web pensando en futuro, para poder realizar cambios desde cualquier parte.

El aumento de complejidad es obvio al intentar migrar su aplicación de escritorio a una solución web, por lo que para dicha solución, servirá como prototipo para una implementación más robusta en el futuro.

Al ser un prototipo no se contará con una base de datos ni un servidor alojado, la información estará disponible mientras la aplicación web se esté ejecutando, pero cuando ésta deje de funcionar toda la información se perderá, pero servirá para simular el funcionamiento real y si es viable a futuro.

La aplicación web en la parte visual, deberá contar con las siguientes funcionalidades:

- Autenticación
- Creación de productos
- Ver listado de productos con su información.
- Eliminación de productos
- Creación de clientes.
- Ver listado de clientes con su información.
- Gráfica de clientes (pie) : Mayores de edad vs menores de edad.
- Gráfica de barras de productos ordenados por precio..

Nota importante: Cada una de las funcionalidades deberá contar con un endpoint o endpoints asociados en el backend, de no cumplir con esto, la funcionalidad no será tomada en cuenta.

Todas las funcionalidades anteriormente descritas deberán estar disponibles en un solo módulo administrador, exceptuando la autenticación.

Lo primero que verá cualquier persona que intente acceder a la aplicación será el módulo de autenticación y hasta que se validen las credenciales, se podrá acceder al módulo administrador.

Autenticación

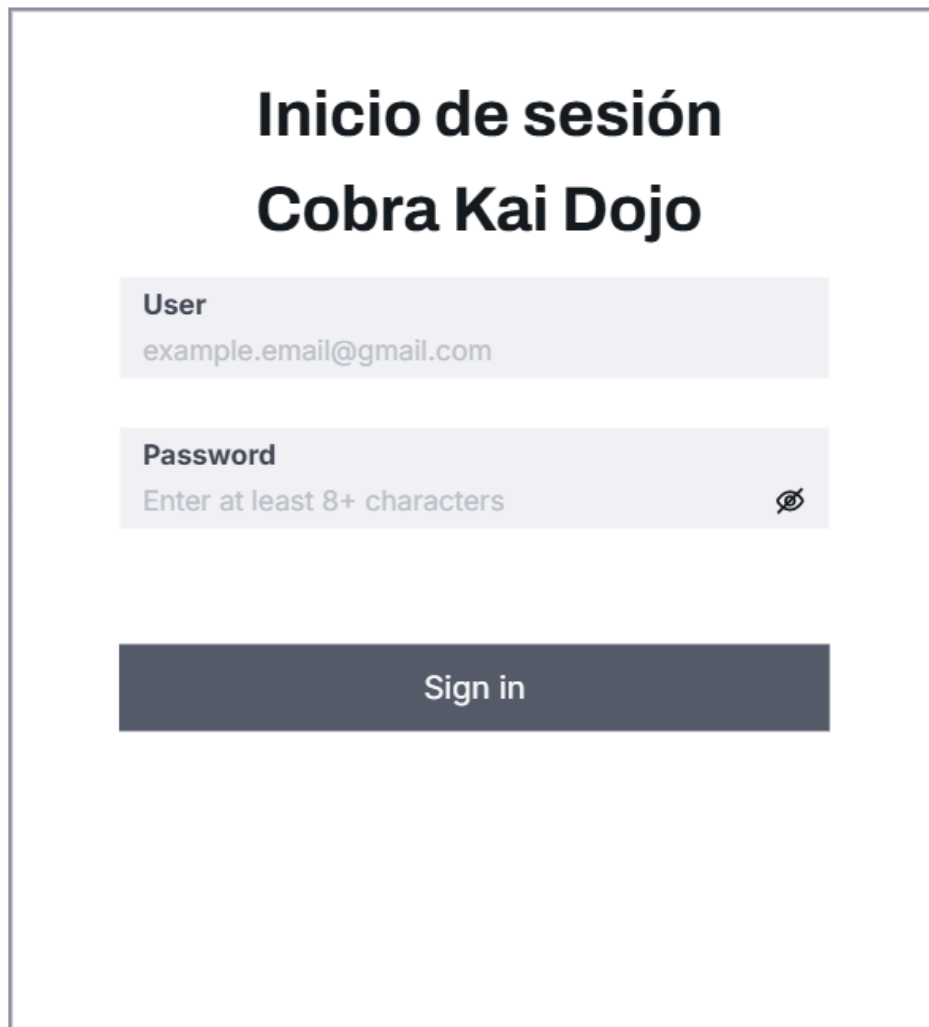
Solamente el sensei podrá acceder al sistema, por lo que se solicita que exista un proceso de autenticación, por lo que si no se ingresan las credenciales correctas, no se permitirá el uso del menú.

Credenciales:

- Usuario: sensei_<número de carnet>
- Contraseña: ipc1_<número de carnet>

Si el usuario o la contraseña son incorrectos deberá mostrar un mensaje de error.


Imagen de referencia del login:



The image shows a login form titled "Inicio de sesión Cobra Kai Dojo". It contains two input fields: "User" with the placeholder text "example.email@gmail.com" and "Password" with the placeholder text "Enter at least 8+ characters" and a toggle icon. Below the fields is a "Sign in" button.

Inicio de sesión
Cobra Kai Dojo

User
example.email@gmail.com

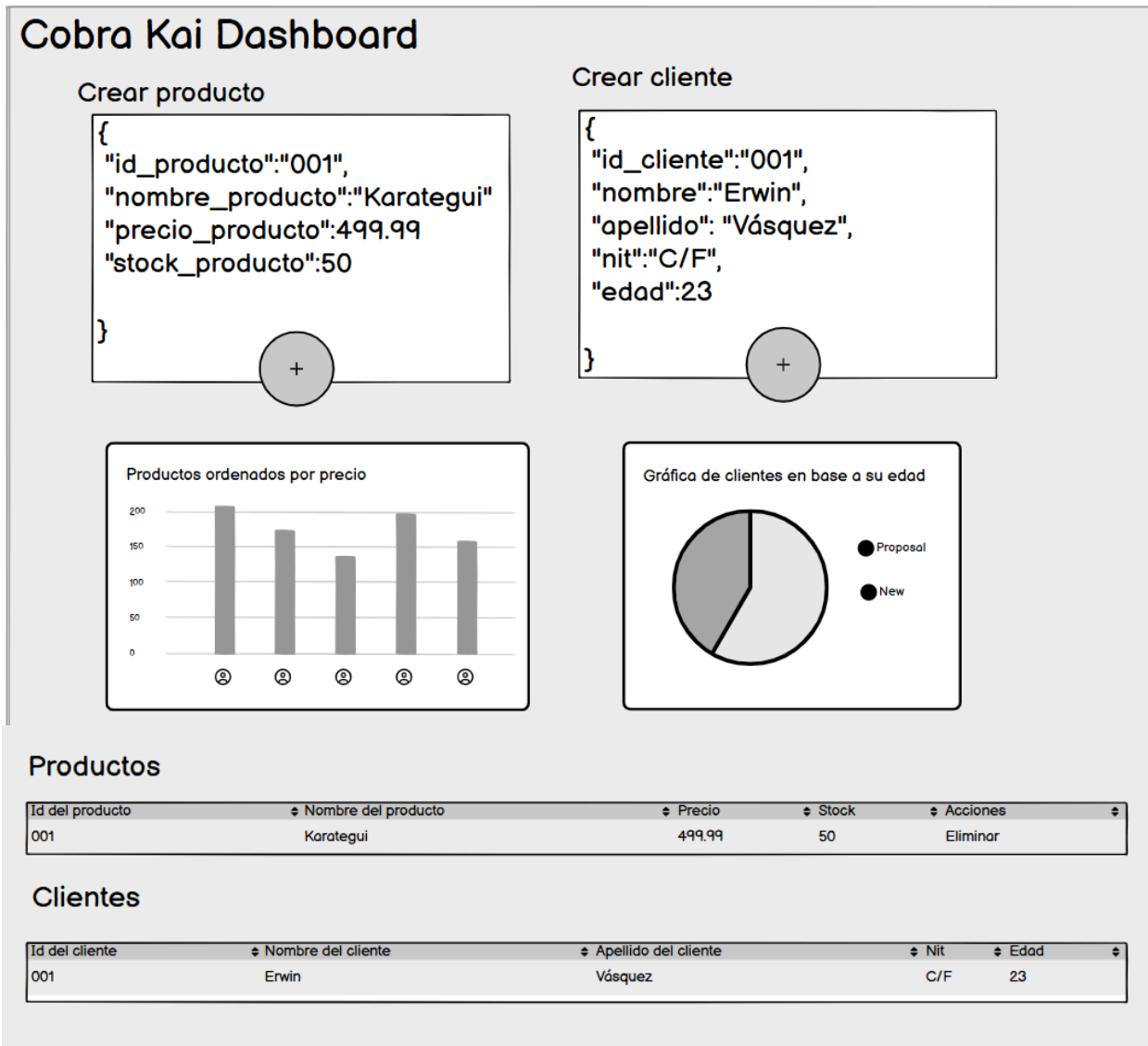
Password
Enter at least 8+ characters 

Sign in

Módulo administrador

Al iniciar sesión de forma correcta, se desplegará el módulo de administración, donde el sensei tendrá diferentes opciones a las cuales se podrá dirigir, todo deberá poder realizarse y ser visible en la misma vista para simular un dashboard.

Al ser un dashboard se deja una imagen de referencia, luego se profundizará en cada punto:



Creación de productos

La aplicación deberá poder registrar productos, estos deben ser enviados en formato JSON al backend para poder validar cada uno de los siguientes casos:

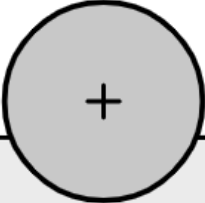
- El precio debe ser mayor a 0.
- El stock puede ser mayor o igual a 0.
- No se puede repetir el producto (Ni el id ni el nombre).

En la interfaz gráfica aparecerá un apartado para escribir texto, en el cual se pueden enviar los datos del producto.

Imagen de referencia:

Crear producto

```
{  
  "id_producto": "001",  
  "nombre_producto": "Karategui",  
  "precio_producto": 499.99,  
  "stock_producto": 50  
}
```



Listado de productos y eliminación

Para poder ver los productos en el sistema, se solicita que se puedan ver todos en una tabla, además cada producto podrá eliminarse en cualquier momento.

Imágen de referencia:

Productos				
Id del producto	Nombre del producto	Precio	Stock	Acciones
001	Karategui	499.99	50	Eliminar

Creación de clientes

La aplicación deberá poder registrar clientes, estos deben ser enviados en formato JSON al backend para poder validar cada uno de los siguientes casos:

- El nit no podrá ir vacío, en dado caso el cliente no posea nit, se debe enviar como "C/F".
- Los id de cliente no se pueden repetir.

Imágen de referencia:

Crear cliente

```
{  
  "id_cliente": "001",  
  "nombre": "Erwin",  
  "apellido": "Vásquez",  
  "nit": "C/F",  
  "edad": 23  
}
```

Listado de clientes

Para poder ver los clientes registrados en el sistema, se solicita que se puedan ver todos en una tabla.

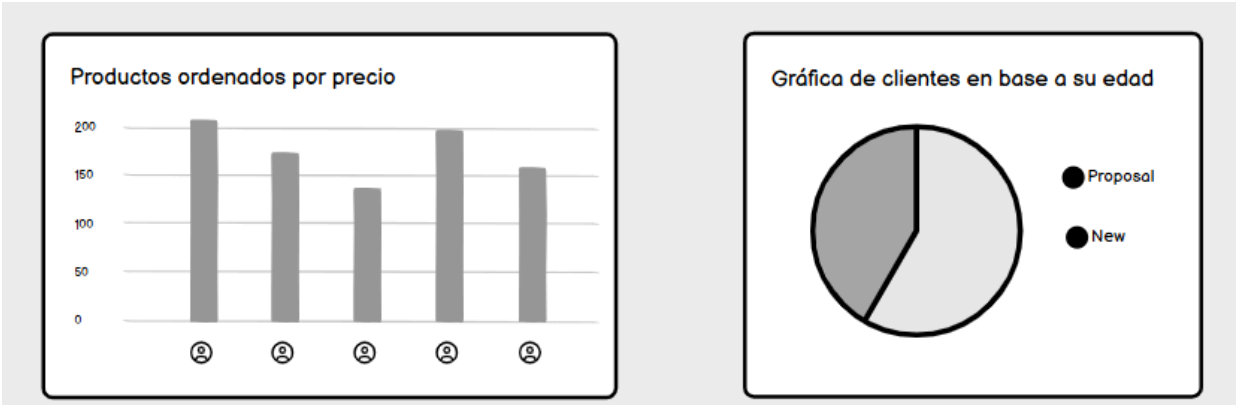
Imágen de referencia:

Clientes				
Id del cliente	Nombre del cliente	Apellido del cliente	Nit	Edad
001	Erwin	Vásquez	C/F	23

Reportes gráficos

El sensei quiere visualizar estadísticas relevantes de sus productos y clientes, para eso se solicitan 2 gráficas:

- Gráfica de barras (productos ordenados por precio descendentemente).
- Gráfica de pie (clientes menores de edad vs mayores de edad).



Consideraciones importantes / restricciones

- Para el frontend se permitirá el uso de React, Vue y Angular como opciones para desarrollarlo.
- El backend debe ser desarrollado obligatoriamente con Express Js en node.
- Cada funcionalidad del frontend debe tener un endpoint o endpoints asociados al backend, de lo contrario la funcionalidad no será tomada en cuenta, recordando que no se tienen que hacer validaciones del lado del frontend.
- Las copias totales o parciales tendrán nota de 0 y serán reportadas a la escuela de Ciencias y Sistemas.
- **Fecha de entrega, jueves 26 de diciembre de 2024, hasta las 23:59:59.**
- Deberá entregarse el link del repositorio privado donde se haya trabajado, el nombre del repositorio es el siguiente: **IPC1_Proyecto02_carnet**. De no entregar el link del repositorio o de no hacer uso de un repositorio no se calificará.
- **No se recibirán entregas tardías**

ENTREGABLES

- ❖ Link del repositorio donde se haya trabajado.
- ❖ El repositorio debe contener documentación, un manual de usuario y un manual técnico en formato pdf.

Penalizaciones

- ❖ De no contar con el uso de un repositorio y no agregar al auxiliar como colaborador se aplicará una penalización.
- ❖ De no contar con documentación de la aplicación se aplicará una penalización.
- ❖ Al no cumplir con alguna restricción o consideración del enunciado se aplicará una penalización (incluso anulación del proyecto).