

ServiApp

Felipe Rojas
Fabian Lopez
Victor Barragan



UNIVERSIDAD
NACIONAL
DE COLOMBIA
SEDE BOGOTÁ



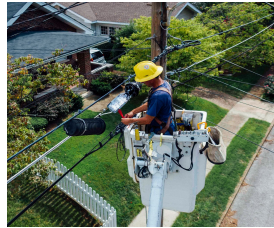
Problema a resolver

- Actualmente tenemos aplicaciones individuales para servicios de mensajería, domicilios y transporte.
- No existe una aplicación que incluya varios servicios técnicos, tradicionales o populares en una misma plataforma.

Cerrajería



Eléctrico doméstico



plomería



- ServiApp será una app web donde se podrán encontrar todo tipo de servicios en la misma plataforma, destacando la facilidad que ofrece este proyecto para la contratación de servicios confiables, seguros y de la mejor calidad.





Requerimientos funcionales

Los requerimientos funcionales básicos correspondientes a esta primera entrega se enlistan a continuación:

- Eliminación de usuarios (Formales y proveedores).
- Selección y contratación de un servicio o proveedor por parte de un usuario final.
- Almacenamiento de los contratos finalizados en forma de historial.

Nota: Más requerimientos funcionales serán agregados en las siguientes entregas.





Requerimientos funcionales

- Inicio de sesión por parte de usuarios registrados a la página web.
- Visualización del listado de proveedores y los servicios disponibles para contratar.



The screenshot shows the ServiaApp login interface. At the top, there is a navigation bar with the ServiaApp logo and three links: 'Inicio', 'Iniciar Sesión', and 'Registrarse como Proveedor de servicios'. Below the navigation bar, there is a large 'Inicia Sesión' button with the ServiaApp logo. To the right of this button, there is a text label 'Inicio de sesion para usuarios registrados' with an arrow pointing to it. Below the 'Inicia Sesión' button, there are input fields for 'Correo' and 'Contraseña', followed by an 'Ingresar' button. At the bottom, there is a link that says '(No tienes una cuenta?)'.



The screenshot shows the 'Registrare' form. It has a title 'Registrare' at the top. Below the title, there are several input fields: 'Tu Nombre completo', 'Numero de contacto', 'Tu email', 'Tu contraseña', 'Tu ciudad', and 'Tipo de servicio' (a dropdown menu). Below these fields is a text area labeled 'Describe tu servicio'. At the bottom of the form is a blue button labeled 'Sign'.

→ Formulario para registrar un nuevo proveedor

- Registro de usuarios (Formales y proveedores)



Uso de estructuras de datos en la solución del problema a resolver

- Implementamos una lista doblemente enlazada para guardar los Clientes y proveedores, esto se hizo teniendo en cuenta que constantemente se está eliminando y agregando nuevos usuarios a la página.

Registrate

Save



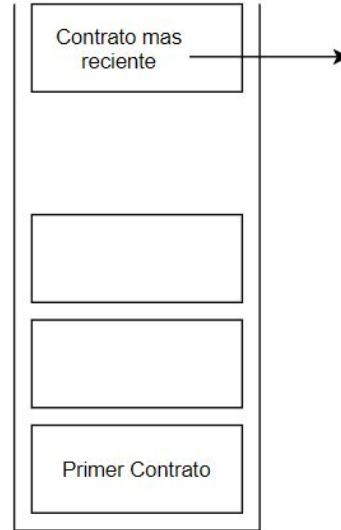
```
let data = {
  id: id,
  firstname: firstName,
  lastname: lastName,
  phoneNumber: phoneNumber,
  city: city,
  email: email,
  score: score,
  description: description,
  owner: owner
};
// los añadimos a la estructura
lista.addToTail(data);
```



Uso de estructuras de datos en la solución del problema a resolver

- Usamos un pila para guardar el historial de contratos ya que nos interesa mostrar los contratos más recientes y al mismo tiempo llevar un control de todos los contratos realizados durante la operación de la página web.
- por último implementamos una lista doblemente enlazada para guardar los proveedores específicos por servicio.

HISTORIAL (Stack)



```
let data = {  
  id: id,  
  usuario: usuario,  
  proveedor: proveedor,  
  estado: estado,  
  pago: pago,  
  cita: cita  
};  
  
pila.push(data);
```

Pruebas y análisis comparativo del uso de las estructuras de datos

Nota: Los datos de prueba se crearon aleatoriamente mediante Faker de node.js

Inicializar la estructura es una de las operaciones más costosas ($O(N^2)$) en nuestras implementaciones. Sin embargo no es una operación muy frecuente en nuestro planteamiento del proyecto.



La eliminación del proveedor supone un costo lineal ($O(N)$). Esta es una operación frecuente en el proyecto y la estructura escogida fue la más óptima luego del análisis de eficiencia



Pruebas y análisis comparativo del uso de las estructuras de datos

Consultar el historial de contratos supone pasar por cada uno de los elementos de la estructura ($O(N)$) sin embargo solo nos interesa consultar los últimos contratos, entonces el costo de esta operación no supone un problema.



Cuando consultamos un servicio es importante poder acceder a toda la lista de proveedores disponibles , el costo de esta operación en la estructura implementada es de ($O(N)$).

