

## SPRINT 4

**Nombre del proyecto:**

G2 Espacios Coworking

### 1. Integrantes de grupo

| NOMBRE INTEGRANTE             | DOCUMENTO     | ROL DEL PROYECTO       |
|-------------------------------|---------------|------------------------|
| Leiner Bracho Ortega          | 77187089      | Gestor Bases de Datos  |
| Miguel Ángel Duncan           | -             | Desarrollador Backend  |
| Diego Armando Moreno          | 1.049.635.723 | Tester                 |
| Eduar Yofree Muñoz            | 1.092.341.532 | Gestor del Proyecto    |
| Juan Camilo Salcedo Rodríguez | 1.014.221.241 | Desarrollador Frontend |

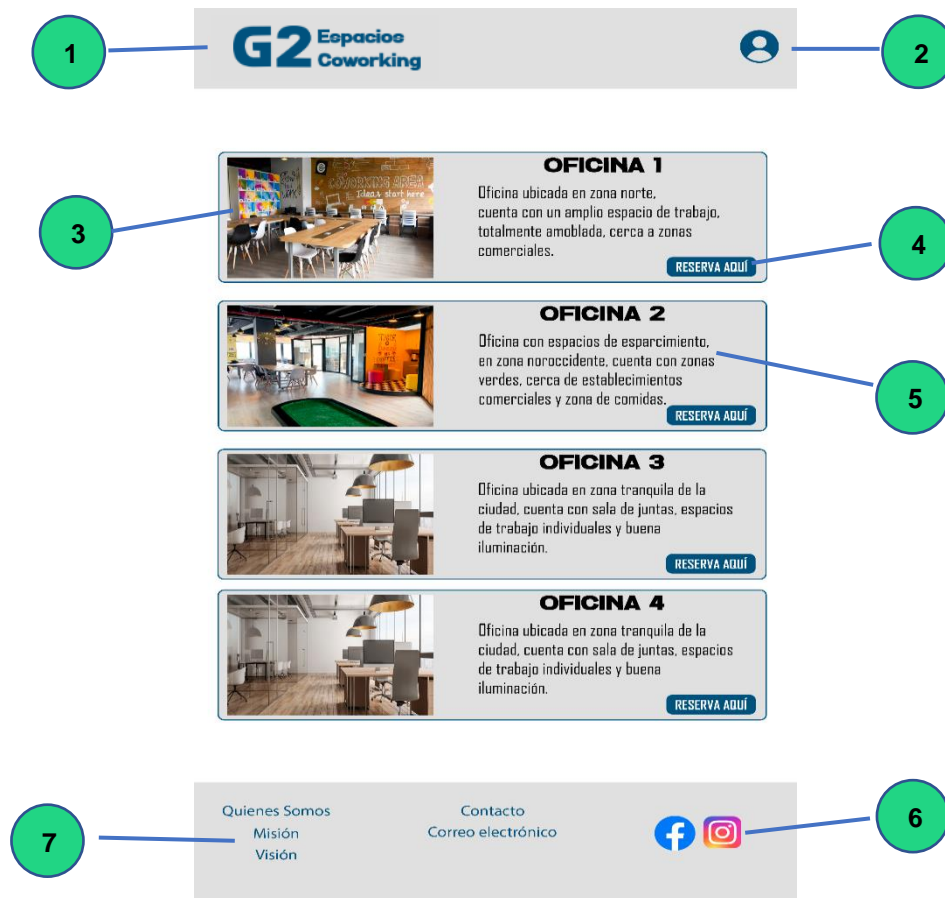
2. Enlace Trello: <https://trello.com/w/plataformacoworking/home>

3. Enlace en la nube del proyecto: <https://demog2.herokuapp.com/>

4. Enlace del repositorio GitHub: <https://github.com/G2MinticUTP/proyecto-coworking.git>

5. Enlace video explicativo de la plataforma con todas sus características y funciones:  
<https://www.youtube.com/watch?v=xh65-fHX22Q>

## 6. Descripción del Front: Página principal.



### IDENTIFICADORES

1. Logo del proyecto.
2. Icono para dirigirse al inicio de sesión.
3. Imagen representativa del espacio coworking.
4. Botón para reservar un espacio coworking.
5. Descripción del espacio coworking.
6. Iconos redes sociales.
7. Información de relevante.

- Front registro de Usuario.



## REGISTRO DE USUARIO

Diagrama del formulario de registro de usuario con los siguientes campos y botones:

- 1. Nombre Completo
- 2. Correo electrónico
- 3. Crear Contraseña
- 4. Confirmar Contraseña
- 5. Teléfono de Contacto
- 6. Usuario (seleccionable)
- 7. Botón Crear Registro

## IDENTIFICADORES

1. Nombre del usuario.
2. Correo electrónico del usuario.
3. Crear una contraseña.
4. Confirmar la contraseña creada.
5. Agregar un teléfono de contacto.
6. Elegir el perfil (Usuario o Administrador).
7. Botón para crear el registro.

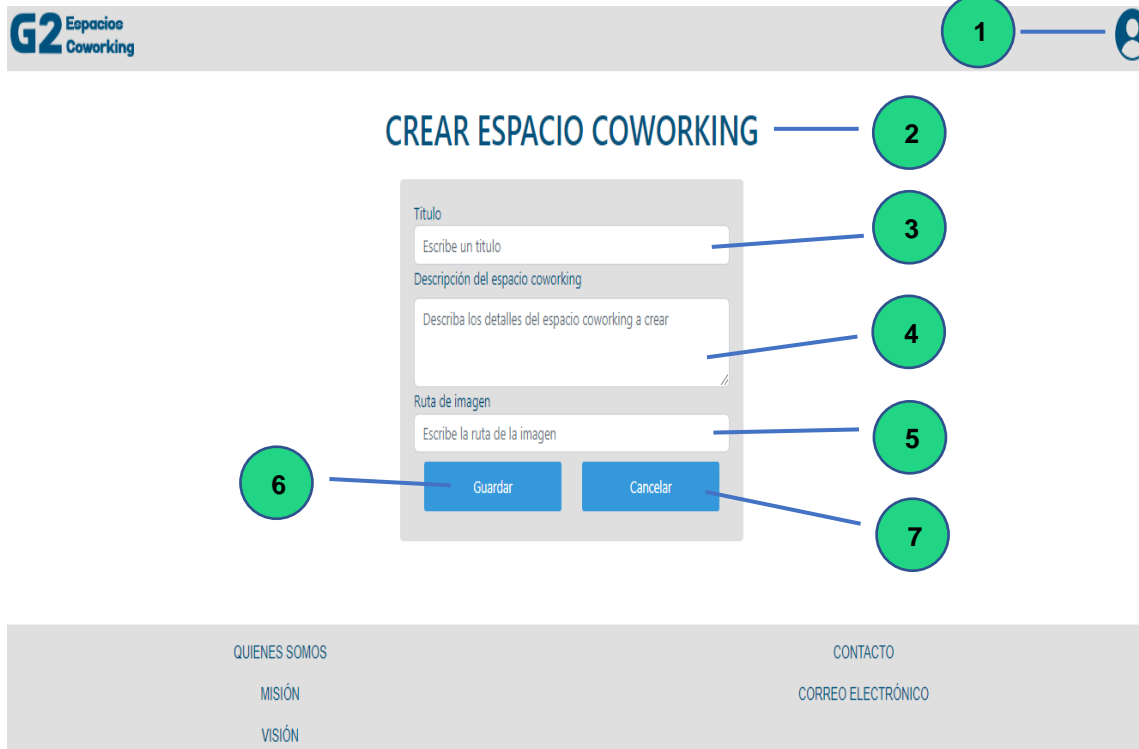
- Front inicio de Sesión.



### IDENTIFICADORES

1. Espacio para ingresar el nombre de usuario.
2. Espacio para ingresar la contraseña del usuario.
3. Botón para iniciar sesión.
4. Enlace para recordar contraseña.
5. Botón para registrar un usuario.

- Front creación de espacios coworking.



**G2 Espacios Coworking**

**1**

**2** CREAR ESPACIO COWORKING

**3** Título  
Escribe un título

**4** Descripción del espacio coworking  
Describe los detalles del espacio coworking a crear

**5** Ruta de imagen  
Escribe la ruta de la imagen

**6** Guardar Cancelar

**7**

QUIENES SOMOS  
MISIÓN  
VISIÓN

CONTACTO  
CORREO ELECTRÓNICO

### IDENTIFICADORES

1. Perfil del usuario.
2. Título del front.
3. Campo para agregar el nombre a un espacio coworking.
4. Campo para diligenciar los detalles de un espacio a crear.
5. Campo para url-imagen del espacio coworking.
6. Botón para aceptar y crear un espacio coworking.
7. Botón para cancelar.

- Front administración de espacios coworking desde el perfil administrador.

**G2 Espacios Coworking**

ADMINISTRACIÓN DE ESPACIOS COWORKING

CREAR ESPACIO

| CÓDIGO | TÍTULO    | DESCRIPCIÓN  | IMAGEN | ACCIONES |
|--------|-----------|--|--------|----------|
| 29     | Espacio 1 | Oficina para 10 personas, cuenta con sala de juntas. |        |          |

QUIENES SOMOS  
MISIÓN  
VISIÓN

CONTACTO  
CORREO ELECTRÓNICO

## IDENTIFICADORES

1. Título del front.
2. Identificador del espacio coworking.
3. Nombre del espacio coworking.
4. Descripción del espacio coworking.
5. Imagen del espacio coworking.
6. Botón para editar el espacio coworking.
7. Botón para eliminar el espacio coworking.
8. Botón para crear un espacio coworking.

- Front actualización y edición de espacios coworking.



## EDITAR ESPACIO COWORKING



1. Título

2. Nombre del espacio coworking

3. Descripción del espacio coworking

4. Ruta de imagen

5. Botón para editar y actualizar el espacio

6. Botón para cancelar la edición del espacio

Formulario de edición de espacio coworking:

Título: Espacio 1

Descripción del espacio coworking: Oficina para 10 personas, cuenta con sala de juntas, con salón de entretenimiento.

Ruta de imagen: <https://amqueretaro.com/wp-content/uploads/2020/11/2-4.jpg>

Botones: Editar, Cancelar

QUIENES SOMOS

MISIÓN

VISIÓN

CONTACTO

CORREO ELECTRÓNICO

### IDENTIFICADORES

1. Título del front.
2. Nombre del espacio coworking.
3. Descripción del espacio coworking.
4. Url de la imagen del espacio coworking.
5. Botón para editar y actualizar el espacio
6. Botón para cancelar la edición del espacio.



- Front formulario de reserva.

**G2 Espacios Coworking**

**1 FORMULARIO DE RESERVA**

Nombre Completo **2**

Correo electrónico **3**

Teléfono de Contacto **4**

Espacio a Reservar **5**

Cantidad de Personas **6**

Tiempo a Reservar **7**

09/09/2022 **8**

**RESERVAR** **9**

**10**

Quienes Somos  
Misión  
Visión

Contacto  
Correo electrónico

f i

## IDENTIFICADORES

1. Título del front.
2. Nombre del usuario que reserva.
3. Correo electrónico del usuario que reserva.
4. Teléfono de contacto del usuario.
5. Nombre de espacio a reservar.
6. Cantidad de personas a usar el espacio.
7. Tiempo que será usado el espacio coworking.
8. Fecha en la cual quiere reservar el espacio coworking.
9. Botón para desplegar un menú.
10. Botón para crear la reserva.



- Front gestión de reservas desde el perfil de usuario.

**G2 Espacios Coworking**

**GESTION DE RESERVAS**

**RESERVAR**

| ESPACIO RESERVADO  | FECHA DE LA RESERVA | CANTIDAD DE PERSONAS | HORA INICIAL DE LA RESERVA | HORA FINAL DE LA RESERVA | PERSONA QUE RESERVO | ACCIONES |  |
|--------------------|---------------------|----------------------|----------------------------|--------------------------|---------------------|----------|--|
| Oficina de trabajo | 2022-10-06          | 6                    | 08:00                      | 17:00                    | Eduar Yofree        |          |  |

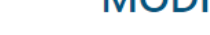
QUIENES SOMOS  
MISIÓN  
VISIÓN

CONTACTO  
CORREO ELECTRÓNICO

## IDENTIFICADORES


1. Título del front.
2. Nombre del espacio coworking.
3. Fecha para la cual se crea la reserva.
4. Cantidad de personas de la reserva.
5. Hora de inicio de la reserva.
6. Hora final de la reserva.
7. Persona quien crea la reserva.
8. Botón para editar la reserva seleccionada.
9. Botón para eliminar la reserva seleccionada.
10. Botón para crear una reserva.

- Front modificación de reservas.



Juan Camilo Salcedo

# MODIFICAR RESERVA



Eduar Yofree

eduardjofree@gmail.com

3209587735

Oficina de trabajo

6

08:00 a. m.

05:00 p. m.

06/10/2022

Modificar reserva

[Ir al Dashboard](#)



Quienes Somos

Misión

Visión

Contacto

Correo electrónico

## IDENTIFICADORES

1. Título del front.
2. Nombre del usuario que reservó un espacio.
3. Correo electrónico del usuario.
4. Teléfono del usuario.
5. Nombre del espacio coworking.
6. Cantidad de personas a usar el espacio.
7. Hora de inicio a usar el espacio.
8. Hora de fin a usar el espacio.
9. Fecha para usar el espacio.
10. Botón para modificar la reserva.
11. Botón para ir a la gestión de reservas.

## 7. Pantallazos del proyecto (Visual Studio Code).

- ICoworkingDao.java (folder acces Dao):

```
1 package com.example.demo.access.dao;      You, hace 2 horas • Proyecto
2
3 import org.springframework.data.jpa.repository.Query;
4 import org.springframework.data.repository.CrudRepository;
5 import com.example.demo.domain.entity.Coworking;
6
7 You, hace 2 horas | 1 author (You)
8
9 public interface ICoworkingDao extends CrudRepository<Coworking, Long>{
10
11     // @Query("select c from Coworking c where c.usuario.id = ?1")
12     // Coworking[] findById(Long idUser);
13
14     @Query("select c from Coworking c where c.id = ?1")
15     Coworking[] findById(Long idCoworking);
16 }
```

- IReservaDao.java (folder acces Dao):

```
1 package com.example.demo.access.dao;      You, hace 2 horas • Proyecto
2
3 import org.springframework.data.jpa.repository.Query;
4 import org.springframework.data.repository.CrudRepository;
5
6 import com.example.demo.domain.entity.Reserva;
7
8 You, hace 2 horas | 1 author (You)
9
10 public interface IReservaDao extends CrudRepository<Reserva, Long> {
11
12     @Query("select c from Reserva c where c.id = ?1")
13     Reserva[] findById(Long idReserva);
14 }
```

- IUserioDao.java (folder acces Dao):

```
1 package com.example.demo.access.dao;
2
3 import org.springframework.data.repository.CrudRepository;
4 import org.springframework.data.jpa.repository.Query;
5
6 //import com.example.demo.domain.entity.Rol;
7 import com.example.demo.domain.entity.Usuario;
8
9
10 You, hace 2 horas | 1 author (You)
11 public interface IUserioDao extends CrudRepository<Usuario, Long>{
12     @Query("select c from Usuario c where c.usuario = ?1 and c.clave = ?2")
13     Usuario findByUsuario(String usuario, String clave);
14 }
15 |
```

- Coworking.java (folder entity):

```
package com.example.demo.domain.entity;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

You, hace 2 horas | 1 author (You)
@Entity
@Table(name = "Coworking")
public class Coworking {

    @Id
    @Column(unique = true, nullable = false)
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String titulo;
    private String descripcion;
    private String media;

    public Coworking() {}

    public Coworking(Long id, String titulo, String descripcion, String media) {
        this.id = id;
        this.titulo = titulo;
        this.descripcion = descripcion;
        this.media = media;
    }

    public Long getId() {
        return id;
    }
}
```

```
public String getTitulo() {
    return titulo;
}

public String getDescripcion() {
    return descripcion;
}

public String getMedia() {
    return media;
}

public void setId(Long id) {
    this.id = id;
}

public void setTitulo(String titulo) {
    this.titulo = titulo;
}

public void setDescripcion(String descripcion) {
    this.descripcion = descripcion;
}

public void setMedia(String media) {
    this.media = media;
}
```

- Reserva.java (folder entity):

```
package com.example.demo.domain.entity;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "Reserva")
public class Reserva {

    @Id
    @Column(unique = true, nullable = false)
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nombres;
    private String correo;
    private String telefono;
    private String espacio;
    private String cantidad;
    private String hora_inicial;
    private String hora_final;
    private String fecha;

    public Reserva() {
    }

    public Reserva(Long id, String nombres, String correo, String telefono, String espacio, String cantidad,
        String hora_inicial, String hora_final, String fecha) {
        this.id = id;
        this.nombres = nombres;
        this.correo = correo;
        this.telefono = telefono;
        this.espacio = espacio;
    }
}
```

```
        this.cantidad = cantidad;
        this.hora_inicial = hora_inicial;
        this.hora_final = hora_final;
        this.fecha = fecha;
    }

    public Long getId() {
        return id;
    }

    public String getNombres() {
        return nombres;
    }

    public String getCorreo() {
        return correo;
    }

    public String getTelefono() {
        return telefono;
    }

    public String getEspacio() {
        return espacio;
    }

    public String getCantidad() {
        return cantidad;
    }

    public String getHora_inicial() {
        return hora_inicial;
    }

    public String getHora_final() {
        return hora_final;
    }
}
```

```
    public String getFecha() {
        return fecha;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public void setNombres(String nombres) {
        this.nombres = nombres;
    }

    public void setCorreo(String correo) {
        this.correo = correo;
    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }

    public void setEspacio(String espacio) {
        this.espacio = espacio;
    }

    public void setCantidad(String cantidad) {
        this.cantidad = cantidad;
    }

    public void setHora_inicial(String hora_inicial) {
        this.hora_inicial = hora_inicial;
    }

    public void setHora_final(String hora_final) {
        this.hora_final = hora_final;
    }

    public void setFecha(String fecha) {
        this.fecha = fecha;
    }
}
```

- Rol.java (folder entity):

```
package com.example.demo.domain.entity;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "Rol")
public class Rol {

    @Id
    @Column(unique = true, nullable = false)
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String descripcion;

    public Rol() {
    }

    public Rol(Long id, String descripcion) {
        this.id = id;
        this.descripcion = descripcion;
    }

    public Long getId() {
        return id;
    }

    public String getDescripcion() {
        return descripcion;
    }
}
```

```
public void setDescripcion(String descripcion) {
    this.descripcion = descripcion;
}

}
```





- Usuario.java (folder entity):

```
package com.example.demo.domain.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Column;

@Entity
@Table(name = "Usuario")
public class Usuario{

    @Id
    @Column(unique = true, nullable = false)
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String usuario;
    private String clave;

    // @JoinColumn(name = "FK_Rol", nullable = false, updatable = false)
    // @ManyToOne(optional = false, cascade = CascadeType.ALL, fetch = FetchType.EAGER)
    // private Rol rol;

    public Usuario() {
    }

    public Usuario(Long id, String usuario, String clave) {
        this.id = id;
        this.usuario = usuario;
        this.clave = clave;
    }
}
```

```
public Long getId() {
    return id;
}

public String getUsuario() {
    return usuario;
}

public String getClave() {
    return clave;
}

public void setId(Long id) {
    this.id = id;
}

public void setUsuario(String usuario) {
    this.usuario = usuario;
}

public void setClave(String clave) {
    this.clave = clave;
}
}
```

- CoworkingService.java (folder service):

```
package com.example.demo.domain.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.example.demo.access.dao.ICoworkingDao;
import com.example.demo.access.dao.IReservaDao;
import com.example.demo.domain.entity.Coworking;
import com.example.demo.domain.entity.Reserva;

You, hace 2 horas | 1 author (You)
@Service
public class CoworkingService implements ICoworkingService{

    @Autowired
    private ICoworkingDao coworkingDao;
    @Autowired
    private IReservaDao reservaDao;

    @Override
    public Coworking[] getCoworkingByUser(Long id) {
        return coworkingDao.findByUserId(id);
    }

    // @Override
    // public Coworking getCoworkingById(Long id) {
    //     return coworkingDao.getCoworkingById(id);
    // }
    //listar todos los espacios de coworking
    @Override
    public Iterable<Coworking> getAllCoworking() {
        return coworkingDao.findAll();
    }
    //lista de coworking por id
    @Override
    public Coworking getById(Long id){
        Coworking coworking = coworkingDao.findById(id).orElse(other; null);
        if (coworking == null) {
```

```
        return new Coworking(id:0L, titulo:"", descripcion:"",media:"");
    }
    return coworking;
}

@Override
public Coworking saveCoworkingById(Coworking coworking){
    return coworkingDao.save(coworking);
}

//crear coworking
@Override
public Coworking saveCoworking(Coworking coworking){
    return coworkingDao.save(coworking);
}

//eliminar coworking
@Override
public void deleteCoworkingById(Long id){
    coworkingDao.deleteById(id);
}

//actualizar coworking
@Override
public Coworking updateCoworking(Coworking coworking){
    return coworkingDao.save(coworking);
}
```

```
//listar todas las reservas de coworking
@Override
public Iterable<Reserva> getAllReservas() {
    return reservaDao.findAll();
}

//listar reservas por id
@Override
public Reserva getByIdReserva(Long id){
    Reserva reserva = reservaDao.findById(id).orElse(other; null);
    if (reserva == null) {
        return new Reserva(id:0L, nombre:"", correo:"",telefono:"",espacios:"",cantidad:"",hora_inicial:"",hora_final:"");
    }
    return reserva;
}

//crear reserva
@Override
public Reserva saveReserva(Reserva reserva){
    return reservaDao.save(reserva);
}

//eliminar reserva
@Override
public void deleteReservaById(Long id){
    reservaDao.deleteById(id);
}

//actualizar reserva
@Override
public Reserva updateReserva(Reserva reserva){
    return reservaDao.save(reserva);
}
```

- ICoworkingService (folder service)

```
1 package com.example.demo.domain.service; You, hace 2 horas
2
3 import com.example.demo.domain.entity.Coworking;
4 import com.example.demo.domain.entity.Reserva;
5
6 You, hace 2 horas | 1 author (You)
7 public interface ICoworkingService {
8     public Coworking[] getCoworkingByUser(Long id);
9     public Iterable<Coworking>getAllCoworking();
10    public Coworking getById(Long id);
11    public Coworking saveCoworkingById(Coworking coworking);
12    public Coworking saveCoworking(Coworking coworking);
13    public void deleteCoworkingById(Long id);
14    public Coworking updateCoworking(Coworking coworking);
15
16    public Iterable<Reserva>getAllReservas();
17    public Reserva getByIdReserva(Long id);
18    public Reserva saveReserva(Reserva reserva);
19    public void deleteReservaById(Long id);
20    public Reserva updateReserva(Reserva coworking);
21
22 }
23
```

- IUserarioService (folder service)

```
1 package com.example.demo.domain.service; You, hace 2 horas
2
3 //import com.example.demo.domain.entity.Rol;
4 import com.example.demo.domain.entity.Usuario;
5
6 You, hace 2 horas | 1 author (You)
7 public interface IUserarioService {
8     public Usuario findByUsuario(String user, String password);
9
10    public Usuario findById(Long id);
11
12 }
```

- UsuarioImplService (folder service)

```
1 package com.example.demo.domain.service;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.stereotype.Service;
5 import com.example.demo.access.dao.IUsuarioDao;
6 import com.example.demo.domain.entity.Usuario;
7
8 You, hace 2 horas | 1 author (You)
9 @Service
10 public class UsuarioImplService implements IUsuarioService{
11
12     @Autowired
13     private IUsuarioDao usuarioDao;
14
15     @Override
16     public Usuario findByUsuario(String usuario, String clave) {
17         Usuario varUser = usuarioDao.findByUsuario(usuario, clave);
18         if (varUser == null) {
19             return new Usuario(id: 0L, usuario: "", clave: "");
20         }
21         return varUser;
22     }
23
24     @Override
25     public Usuario findById(Long id) {
26         Usuario user = usuarioDao.findById(id).orElse(other: null);
27         if (user == null) {
28             return new Usuario(id: 0L, usuario: "", clave: "");
29         }
30         return user;
31     }
32 }
```

- ReservaController.java (folder presentation):

```
1 package com.example.demo.presentation;
2
3 You, hace 2 horas | 1 author (You)
4 public class ReservaController {
5
6 }
```

- UsuarioController (folder controller)

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import com.example.demo.domain.entity.Coworking;
import com.example.demo.domain.entity.Reserva;
import com.example.demo.domain.entity.Usuario;
import com.example.demo.domain.service.ICoworkingService;
import com.example.demo.domain.service.IUsuarioService;

You, hace 2 horas | 1 author (You)
@Controller
public class UsuarioController {

    @Autowired
    private IUsuarioService usuarioService;

    @Autowired
    private ICoworkingService coworkingService;

    @Autowired
    private ICoworkingService reservaService;
```

```
Long idlogueado;

// listar Front Página principal
@GetMapping("/{}/home")
public String home(Model model) {
    model.addAttribute(attributeName: "AllCoworking", coworkingService.getAllCoworking());
    return "home"; //va hacia el template
}

// listar espacios de coworking
@GetMapping("/{}/espaciosCoworking")
public String espaciosCoworking(Model model) {
    model.addAttribute(attributeName: "AllCoworking", coworkingService.getAllCoworking());
    return "espaciosCoworking"; //va hacia el template
}

// Front Inicio de Sesión
@GetMapping("/InicioSesion")
public String oAuth(Model model) {
    You, hace 2 horas • Proyecto final G2Coworking
    model.addAttribute(attributeName: "usuariologueado", new Usuario());
    return "InicioSesion";
}

//procesar usuario y contraseña
@PostMapping("/IniciarSesion")
public String iniciarSesion(@ModelAttribute("usuariologueado") Usuario user,
    Model model) {
    Usuario usuario = usuarioService.findbyUsuario(user.getUsuario()),
    user.getClave();
    String pg = "";
}
```

```
if (usuario.getId() != 0) {
    idlogueado = usuario.getId();
    model.addAttribute(attributeName: "idUsuario", idlogueado);
    model.addAttribute(attributeName: "usuario", usuario.getUsuario());
    if (usuario.getId() == 1) {
        pg = "redirect:/espaciosCoworking";
    }
    else {
        pg = "redirect:/gestionReservas";
    }
}
else {
    pg = "credenciales_incorrectas";
}
return pg;
}
```



```
// Front Recuperar Contraseña
@GetMapping("/RecuperarContraseña")
public String RecuperarContraseña(Model model) {
    Coworking coworking = new Coworking();
    model.addAttribute(attributeName: "coworking", coworking);
    return "RecuperarContraseña";
}

// Front Registro Usuario
@GetMapping("/RegistroUsuario")
public String RegistroUsuario(Model model) {
    Coworking coworking = new Coworking();
    model.addAttribute(attributeName: "coworking", coworking);
    return "RegistroUsuario";
}

// Front Formulario de reserva
@GetMapping("/FormularioReserva")
public String FormularioReserva(Model model) {
    Coworking coworking = new Coworking();
    model.addAttribute(attributeName: "coworking", coworking);
    return "FormularioReserva";
}

// crear espacio de coworking
@GetMapping("/crearCoworking")
public String crearCoworking(Model model) {
    Coworking coworking = new Coworking();
    model.addAttribute(attributeName: "coworking", coworking);
    return "crearCoworking";
}

@PostMapping("/guardar")
public String saveCoworking(@ModelAttribute("coworking") Coworking coworking) {
    coworkingService.saveCoworking(coworking);

    return "redirect:/espaciosCoworking";
}
```

```
// editar espacio de coworking
@GetMapping("/editarCoworking/{id}")
public String actualizarCoworking(@PathVariable Long id, Model model) {
    Coworking coworking = coworkingService.getById(id);
    if (coworking.getId() != 0) {
        model.addAttribute(attributeName: "coworking", coworking);
        return "editarCoworking";
    } else {
        return "editarCoworking";
    }
}

@PostMapping("/editar/{id}")
public String updateCoworking(@PathVariable Long id,
    @ModelAttribute("coworking") Coworking coworking,
    Model model) {

    Coworking coworkingExistente = coworkingService.getById(id);

    coworkingExistente.setId(coworking.getId());
    coworkingExistente.setTitulo(coworking.getTitulo());
    coworkingExistente.setDescripcion(coworking.getDescripcion());

    coworkingService.saveCoworking(coworking);
    return "redirect:/espaciosCoworking";
}

//eliminar espacio de coworking
@PostMapping("/eliminar/{id}")
public String deleteCoworking(@PathVariable Long id, Model model) {
    coworkingService.deleteCoworkingById(id);

    return "redirect:/espaciosCoworking";
}
```

```
//gestionReservas
@GetMapping("/gestionReservas")
public String gestionReservas(Model model) {
    model.addAttribute(attributeName: "AllReservas", reservaService.getAllReservas());
    return "gestionReservas"; //va hacia el template
}

//reserva con id
@GetMapping("/reserva/{id}")
public String reserva(@PathVariable Long id, Model model) {
    Reserva reserva = reservaService.getIdReserva(id);
    if (reserva.getId() != 0) {
        model.addAttribute(attributeName: "reserva", reserva);
        return "reserva";
    } else {
        return "reserva";
    }
}

// crear espacio de coworking
@GetMapping("/crearReserva")
public String crearReserva(Model model) {
    Reserva coworking = new Reserva();
    model.addAttribute(attributeName: "reserva", coworking);
    return "crearReserva";
}

@PostMapping("/guardarReserva")
public String saveReserva(@ModelAttribute("reserva") Reserva reserva) {
    reservaService.saveReserva(reserva);

    return "redirect:/gestionReservas";
}
```

```
// editar reserva
@GetMapping("/editarReserva/{id}")
public String editarReserva(@PathVariable Long id, Model model) {
    Reserva reserva = reservaService.getByIdReserva(id);
    if (reserva.getId() != 0) {
        model.addAttribute(attributeName: "reserva", reserva);
        return "editarReserva";
    } else {
        return "editarReserva";
    }
}

@PostMapping("/editarReserva/{id}")
public String updateReserva(@PathVariable Long id,
    @ModelAttribute("reserva") Reserva reserva,
    Model model) {

    Reserva reservaExistente = reservaService.getByIdReserva(id);

    reservaExistente.setId(reserva.getId());
    reservaExistente.setNombres(reserva.getNombres());
    reservaExistente.setCorreo(reserva.getCorreo());
    reservaExistente.setTelefono(reserva.getTelefono());
    reservaExistente.setEspacio(reserva.getEspacio());
    reservaExistente.setCantidad(reserva.getCantidad());
    reservaExistente.setHora_inicial(reserva.getHora_inicial());
    reservaExistente.setHora_final(reserva.getHora_final());
    reservaExistente.setFecha(reserva.getFecha());

    reservaService.saveReserva(reserva);
    return "redirect:/gestionReservas";
}
```

```
//eliminar espacio de coworking
@PostMapping("eliminarReserva/{id}")
public String deleteReserva(@PathVariable Long id, Model model) {
    reservaService.deleteReservaById(id);

    return "redirect:/gestionReservas";
}
}
```

- UsuarioRepository (folder repositories)

```
1 package com.example.demo.repositories;
2
3 public class UsuarioRepository {
4
5 }
6
```





- DemoApplication

```
1 package com.example.demo;           You, hace 2 horas • Proyecto final G2
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 You, hace 2 horas | 1 author (You)
7 @SpringBootApplication
8 public class DemoApplication {
9
10     Run | Debug
11     public static void main(String[] args) {
12         SpringApplication.run(DemoApplication.class, args);
13     }
14 }
```

- ServletInitializer

```
1 package com.example.demo;           You, hace 2 horas • Proyecto final G2Coworking ...
2
3 import org.springframework.boot.builder.SpringApplicationBuilder;
4 import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
5
6 You, hace 2 horas | 1 author (You)
7 public class ServletInitializer extends SpringBootServletInitializer {
8
9     @Override
10     protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
11         return application.sources(DemoApplication.class);
12     }
13 }
14
```

- Estilos (folder resources – static - css)

```
/*Estilos pagina principal*/
header {
  margin-bottom: 2rem;
}

.logo {
  margin-left: 1.5rem;
}

.titulos {
  text-align: center;
  margin-bottom: 1.5rem;
  color: #04537c;
}

.cards {
  border-color: #04537c;
  height: 230px;
  width: 1100px;
  margin: auto;
}

.cards:hover {
  box-shadow: 0px 0px 11px 0px #04537c;
  cursor: auto;
}

.cards img {
  height: 210px;
  width: 340px;
  margin-top: 0.5rem;
}
```

```
/*Códigos para footer pagina principal*/
.footer {
  background-color: #fdfdfd;
  color: #04537c;
  margin-top: 5rem;
  font-family: sans-serif;
  font-size: medium;
}

/*Estilos para formularios*/
* {
  box-sizing: border-box;
}

*:focus {
  outline: none;
}

.login {
  margin: 10px auto 50px;
  width: 400px;
}

.login-registro {
  margin: 10px auto 10px;
  width: 500px;
}

.login-screen {
  background-color: #fdfdfd;
  padding: 20px;
  border-radius: 5px;
}
```

```
.login-screen-recuperacion {
  background-color: #fdfdfd;
  padding: 20px;
  margin-bottom: 25rem;
  border-radius: 5px;
}

.app-title {
  text-align: center;
  color: #7777;
}

.login-form {
  text-align: center;
  margin-top: 1rem;
}

.control-group {
  margin-bottom: 10px;
}

input {
  text-align: center;
  background-color: #white;
  border: 2px solid transparent;
  border-radius: 3px;
  font-size: 16px;
  font-weight: 200;
  padding: 10px 0;
  width: 250px;
  transition: border 0.5s;
}

input:focus {
  border: 2px solid #3498db;
  box-shadow: none;
}
```

```
.select-perfil {
  text-align: center;
  background-color: #white;
  border: 2px solid transparent;
  border-radius: 3px;
  font-size: 16px;
  font-weight: 200;
  padding: 10px 0;
  width: 250px;
  transition: border 0.5s;
}

.cantidad-personas {
  text-align: center;
  background-color: #white;
  border: 2px solid transparent;
  border-radius: 3px;
  font-size: 16px;
  font-weight: 200;
  padding: 10px 0;
  width: 250px;
  transition: border 0.5s;
}

.calendario {
  text-align: center;
  background-color: #white;
  border: 2px solid transparent;
  border-radius: 3px;
  font-size: 16px;
  font-weight: 200;
  padding: 10px 0;
  width: 250px;
  transition: border 0.5s;
  margin: auto auto 0.7rem;
}
```

```
.btn {
  border: 2px solid transparent;
  background: #3498db;
  color: #ffffff;
  font-size: 16px;
  line-height: 25px;
  padding: 10px 0;
  text-decoration: none;
  text-shadow: none;
  border-radius: 3px;
  box-shadow: none;
  transition: 0.25s;
  display: block;
  width: 250px;
  margin: 0 auto;
}

.btn:hover {
  background-color: #2980b9;
}

.login-link {
  font-size: 15px;
  color: #444;
  display: block;
  margin-top: 12px;
}

.form-control {
  text-align: left;
}

.subir-archivo .text-area {
  color: #04537c;
  text-align: left;
}
```

```
.subir-archivo .file {
  color: #04537c;
  background-color: #e6e6e6;
  margin-top: 1rem;
}

.btn-enviar {
  border: 2px solid transparent;
  background: #3498db;
  color: #ffffff;
  font-size: 16px;
  padding: 10px 0;
  border-radius: 3px;
  box-shadow: none;
  transition: 0.25s;
  display: block;
  width: 200px;
  margin: 0.5rem auto;
}

.btn-enviar:hover {
  box-shadow: 0px 0px 11px 0px #04537c;
}
```

```
width: 250px;
margin: 1rem auto;
}

.btn-cancelar {
  border: 2px solid transparent;
  background: #e74c3c;
  color: #ffffff;
  font-size: 16px;
  line-height: 25px;
  padding: 10px 0;
  text-decoration: none;
  text-shadow: none;
  border-radius: 3px;
  box-shadow: none;
  transition: 0.25s;
  display: block;
  width: 250px;
  margin: 1rem auto;
}

.boton-crear-espacios {
  margin-top: 2rem;
  text-decoration: none;
  border: 2px solid transparent;
  background: #3498db;
  color: #ffffff;
  font-size: 16px;
  padding: 10px 10px;
  border-radius: 3px;
  box-shadow: none;
  transition: 0.25s;
  width: 100px;
}
```

```
/*Códigos para tabla administración espacios*/
#tabla {
  margin-left: auto;
  margin-right: auto;
  margin-top: 3rem;
  border-style: groove;
  margin-bottom: 10.3rem;
}

th {
  background-color: #fdfdfd;
  color: #04537c;
  padding: 20px;
  border: 2px solid #black !important;
  border-width: 2px;
  text-align: center;
}

tr,
td {
  padding: 20px;
  border: 2px solid #black !important;
  border-width: 2px;
  text-align: center;
}

imgCoworking {
  height: 180px;
}

.btn-primary-admin {
  width: 100px;
}

.eliminar {
  background: #fff;
  border: none;
}
```

```
.labelCoworking {
  color: #04537c;
}

imgCoworking {
  width: 80%;
}

.boton {
  display: flex;
}
```



- Template Home (folder resource - templates)

```

1  <!DOCTYPE html>      You, hace 3 horas • Proyecto final G2Coworking ...
2  <html xmlns:th="http://www.thymeleaf.org">
3      <head>
4          <meta charset="UTF-8" />
5          <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6          <meta
7              name="viewport"
8              content="width=device-width, initial-scale=1.0"
9          />
10         <title>G2 Coworking</title>
11         <link
12             href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
13             rel="stylesheet"
14             integrity="sha384-gh2yIJJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvAPhHn1/vI1Bx"
15             crossorigin="anonymous"
16         />
17         <link th:href="@{css/G2styles.css}" rel="stylesheet" />
18     </head>
19     <body>
20         <!--Etiquetas para navbar-->
21         <header>
22             <nav class="navbar" style="background-color: #dfdfdf">
23                 <div class="container-fluid">
24                     <a class="navbar-brand" href="#">
25                         
30                     </a>
31                     <a class="nav-link" th:href="@{/InicioSesion}">
32                         </a>
33                 </div>
34             </nav>
35         </header>

```

- Template crearCoworking.html (folder resource - templates)

```

1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3      <head>
4          <meta charset="UTF-8" />
5          <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6          <meta
7              name="viewport"
8              content="width=device-width, initial-scale=1.0"
9          />
10         <title>G2 Crear Espacio</title>
11         <link
12             href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
13             rel="stylesheet"
14             integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHhNl/vI1Bx"
15             crossorigin="anonymous"
16         />
17         <link th:href="@{css/G2styles.css}" rel="stylesheet" />
18     </head>
19     <body>
20         <!--Etiquetas para navbar-->      You, hace 3 horas • Proyecto final G2Coworking ...
21         <header>
22             <nav class="navbar" style="background-color: #dfdfdf">
23                 <div class="container-fluid">
24                     <a class="navbar-brand" th:href="@{/Home}">
25                     </a>
26                     <a class="nav-link" th:href="@{/InicioSesion}">
27                     </a>
28                 </div>
29             </nav>
30         </header>
31
32         <div class="títulos">
33             <h1>CREAR ESPACIO COWORKING</h1>
34         </div>
35
36         <!--Etiquetas para el Login-->

```

- Template crearReserva.html (folder resource - templates)

```

1 <!DOCTYPE html> You, hace 3 horas • Proyecto final G2Coworking ...
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta
7       name="viewport"
8       content="width=device-width, initial-scale=1.0"
9     />
10    <title>G2 Formulario de Reserva</title>
11    <link
12      href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
13      rel="stylesheet"
14      integrity="sha384-gH2yIJqKdNHPEq0n4MqA/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvAphHhNl/vI1Bx"
15      crossorigin="anonymous"
16    />
17    <link th:href="@{css/G2styles.css}" rel="stylesheet" />
18  </head>
19  <body>
20    <!--Etiquetas para navbar-->
21    <header>
22      <nav class="navbar" style="background-color: #dfdfdf">
23        <div class="container-fluid">
24          <a class="navbar-brand" th:href="@{/Home}"></a>
25          <a class="nav-link" th:href="@{/InicioSesion}">
26            </a>
27        </div>
28      </nav>
29    </header>
30
31    <div class="títulos">
32      <h1>CREAR RESERVA</h1>
33    </div>
34
35    <!--Etiquetas para el Login-->

```

- Template credenciales\_incorrectas.html (folder resource - templates)

```

1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3
4  <head>
5      <meta charset="utf-8">
6      <title>Mensaje</title>
7      <meta name="viewport" content="width=device-width, initial-scale=1">
8      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"
9          integrity="sha384-iYQeCzEYFbkjA/T2uDLTPkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">
10     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
11         integrity="sha384-u10knCvxwV5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvzbZlHgTPOOmMi466C8"
12         crossorigin="anonymous"></script>
13     <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
14         integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfExiDAYTujMAeBAsjFuCZSmKbSSunQlMh/jp3"
15         crossorigin="anonymous"></script>
16     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.min.js"
17         integrity="sha384-7VPbUDKoPSGfNtYi0QogXtr74QeVeeIs99Qfg5YCF+TidwNdjvaKZx19NZ/e6oz"
18         crossorigin="anonymous"></script>
19 </head>
20
21 <body class="text-bg-secondary m-0 vh-100 row justify-content-center align-items-center">
22     <div class="card text-bg-dark" style="width: 25rem;">
23         <div class="card-body text-center">
24             <h2 class="card-title">Credenciales incorrectas</h2>
25             <div class="col-12">
26                 <a th:href="@{/InicioSesion}" class="btn btn-primary btn-sm mb-3" aria-current="page">Volver</a>
27             </div>
28         </div>
29     </div>
30
31 </body>
32
33 </html>

```

- Template editarCoworking.html (folder resource - templates)

```

1 <!DOCTYPE html> You, hace 3 horas • Proyecto final G2Coworking ...
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
7     <title>G2 Editor Espacio</title>
8     <link th:href="@{/css/G2styles.css}" rel="stylesheet" />
9     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-gl
10   </head>
11   <body>
12     <!--Etiquetas para navbar-->
13     <header>
14       <nav class="navbar" style="background-color: #dfdfdf">
15         <div class="container-fluid">
16           <a class="navbar-brand" th:href="@{/Home}">
17           </a>
18           <a class="nav-link" th:href="@{/InicioSesion}">
19             </a>
20         </div>
21       </nav>
22     </header>
23
24     <div class="títulos">
25       <h1>EDITAR ESPACIO COWORKING</h1>
26     </div>
27
28     <!--Etiquetas para el Login-->
29     <div class="login-registro">
30       <div class="login-screen">
31         <div class="subir-archivo">
32           <form th:action="@{/editar/{id}(id=${coworking.id})}" th:object="${coworking}" method="post">
33             <div class="text-area">
34               <label class="tituloCoworking">Titulo</label>
35               <input type="text" class="form-control" id="id_coworking" placeholder="Escriba un título" th:field="${tit

```



- Template editarReserva.html (folder resource - templates)

```

1  <!DOCTYPE html>      You, hace 3 horas • Proyecto final G2Coworking ...
2  <html xmlns:th="http://www.thymeleaf.org">
3      <head>
4          <meta charset="UTF-8" />
5          <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6          <meta
7              name="viewport"
8              content="width=device-width, initial-scale=1.0"
9          />
10         <title>G2 Modificar Reserva</title>
11         <link
12             href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
13             rel="stylesheet"
14             integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvAphHNNL/vI1Bx"
15             crossorigin="anonymous"
16         />
17         <link th:href="@{/css/G2styles.css}" rel="stylesheet" />
18     </head>
19     <body>
20         <!--Etiquetas para navbar-->
21         <header>
22             <nav class="navbar" style="background-color: #dfdfdf">
23                 <div class="container-fluid">
24                     <a class="navbar-brand" th:href="@{/Home}"></a>
25                     <a class="nav-link" th:href="@{/InicioSesion}">
26                         </a>
27                 </div>
28             </nav>
29         </header>
30
31         <div class="títulos">
32             <h1>MODIFICAR RESERVA</h1>
33         </div>
34
35         <!--Etiquetas para el Login-->
36         <div class="Login-parciales">

```

- Template espaciosCoworking.html (folder resource - templates)

```

1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3      <head>
4          <meta charset="utf-8" />
5          <title>G2 Administración Coworking</title>
6          <meta
7              name="viewport"
8              content="width=device-width, initial-scale=1"
9          />
10         <!--<link rel="stylesheet" type="text/css" href="G2styles.css" />-->
11         <link th:href="@{css/G2styles.css}" rel="stylesheet" />
12         <link
13             href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
14             rel="stylesheet"
15             integrity="sha384-gH2yIJqKdNHPEq0n4Mqqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx"
16             crossorigin="anonymous"
17         />
18     </head>
19
20     <body>
21         <!--Etiquetas para navbar-->
22         <header>
23             <nav class="navbar" style="background-color: #d9d9d9">
24                 <div class="container-fluid">
25                     <a
26                         class="navbar-brand"
27                         th:href="@{/Home}"
28                     >
33                     </a>
34                     <a class="nav-link" th:href="@{/InicioSesion}">
35                         </a>

```

- Template gestionReservas.html (folder resource - templates)

```

1  <!DOCTYPE html>      You, hace 3 horas • Proyecto final G2Coworking ...
2  <html xmlns:th="http://www.thymeleaf.org">
3
4  <head>
5      <meta charset="utf-8" />
6      <title>G2 Administración Coworking</title>
7      <meta name="viewport" content="width=device-width, initial-scale=1" />
8      <!--<link rel="stylesheet" type="text/css" href="G2styles.css" />-->
9      <link th:href="@{css/G2styles.css}" rel="stylesheet" />
10     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet"
11         integrity="sha384-gh2yIjQKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvAphHnL/vI1Bx" crossorigin="anonymous" />
12 </head>
13
14 <body>
15     <!--Etiquetas para navbar-->
16     <header>
17         <nav class="navbar" style="background-color: #dfdfdf">
18             <div class="container-fluid">
19                 <a class="navbar-brand" th:href="@{/Home}">
20                 </a>
21                 <a class="nav-link" th:href="@{/InicioSesion}">
22                     </a>
23             </div>
24         </nav>
25     </header>
26
27     <div class="títulos">
28         <h2>GESTION DE RESERVAS</h2>
29         <br />
30         <a class="boton-crear-espacios" th:href="@{/Home}">RESERVAR</a>
31     </div>
32
33     <!--Etiquetas para tabla-->
34     <div>
35         <table id="tabla">
36             <thead>

```

- Template InicioSesion.html (folder resource - templates)

```
1 <!DOCTYPE html> You, hace 3 horas • Proyecto final G2Coworking ...
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>G2 Inicio de Sesión</title>
8     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet"
9       integrity="sha384-gH2yIJqKdNHPEq0n4MqA/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApPHN1/vI1Bx" crossorigin="anonymous" />
10    <link th:href="@{css/G2styles.css}" rel="stylesheet" />
11  </head>
12  <body>
13    <!--Etiquetas para navbar-->
14    <header>
15      <nav class="navbar" style="background-color: #dfdfdf">
16        <div class="container-fluid">
17          <a class="navbar-brand" th:href="@{/Home}">
18            
19          </a>
20          <a class="nav-link" th:href="@{/InicioSesion}">
21            </a>
22        </div>
23      </nav>
24    </header>
25
26    <div class="títulos">
27      <h1>INICIO DE SESIÓN</h1>
28    </div>
29
30    <!--Etiquetas para el Login-->
31    <div class="login">
32      <div class="login-screen">
33        <div class="app-title">
34          
35        </div>
36      </div>
37    </div>
```

- Template RecuperarContraseña.html (folder resource - templates)

```

1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3      <head>
4          <meta charset="UTF-8" />
5          <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6          <meta
7              name="viewport"
8              content="width=device-width, initial-scale=1.0"
9          />
10         <title>G2 Recuperación de Contraseña</title>
11         <link
12             href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
13             rel="stylesheet"
14             integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvAphHnL/vI1Bx"
15             crossorigin="anonymous"
16         />
17         <link th:href="@{css/G2styles.css}" rel="stylesheet" />
18     </head>
19     <body>
20         <!--Etiquetas para navbar-->
21         <header>
22             <nav class="navbar" style="background-color: #dfdfff">
23                 <div class="container-fluid">
24                     <a class="navbar-brand" th:href="@{/Home}">
25                         >
26                         
31                     </a>
32                     <a class="nav-link" th:href="@{/InicioSesion}">
33                         </a>
34                 </div>
35             </nav>
36         </header>

```

- Template RegistroUsuario.html (folder resource - templates)

```

1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3      <head>
4          <meta charset="UTF-8" />
5          <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6          <meta
7              name="viewport"
8              content="width=device-width, initial-scale=1.0"
9          />
10         <title>G2 Registro de Usuario</title>
11         <link
12             href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
13             rel="stylesheet"
14             integrity="sha384-gH2yIjQKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx"
15             crossorigin="anonymous"
16         />
17         <link th:href="@{css/G2styles.css}" rel="stylesheet" />
18     </head>
19     <body>
20         <!--Etiquetas para navbar-->
21         <header>
22             <nav class="navbar" style="background-color: #dfdfdf">
23                 <div class="container-fluid">
24                     <a class="navbar-brand" th:href="@{/Home}">
25                         
30                     </a>
31                     <a class="nav-link" th:href="@{/InicioSesion}">
32                         </a>
33                 </div>
34             </nav>
35         </header>
36     </body>

```

- Application.properties

```
1  spring.application.name=DEMO
2  #server.port=8001
3
4  #Data source
5  #Indica el driver/lib para conectar java a mysql
6  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
7
8  #Url donde esta el servicio de tu mysql y el nombre de la base de datos
9  #datos locales
10 spring.datasource.url=jdbc:mysql://us-cdbr-east-06.cleardb.net:3306/heroku_b6b386f98a95cb0?user=b33d9ffadec675&password=0ad48f9f
11
12 #Usuario y contraseña para tu base de datos descrita en la linea anterior
13 #datos locales
14 spring.datasource.username=b33d9ffadec675
15 spring.datasource.password=0ad48f9f
16 #[opcional]Imprime en tu consola las instrucciones hechas en tu base de datos.
17 spring.jpa.show-sql = true
18 spring.jpa.hibernate.ddl-auto=update
19
20 #-----datos en la nube-----
21 #spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
22 #spring.datasource.url=jdbc:mysql://us-cdbr-east-06.cleardb.net:3306/heroku_6cae2836ab8c42a?user=b8ccbe1372ae90&password=b5d62807
23 #spring.datasource.username=b8ccbe1372ae90
24 #spring.datasource.password=b5d62807
25 #spring.jpa.show-sql = true
26 #spring.jpa.hibernate.ddl-auto=update
27 #mysql://b8ccbe1372ae90:b5d62807@us-cdbr-east-06.cleardb.net/heroku_6cae2836ab8c42a? clave base de datos con xampp erebus1988
28 #mysql://b33d9ffadec675:0ad48f9f@us-cdbr-east-06.cleardb.net/heroku_b6b386f98a95cb0?reconnect=true
```



- Base de datos phpMyAdmin

|                          | # | Nombre      | Tipo         | Cotejamiento       | Atributos | Nulo | Predeterminado | Comentarios | Extra          | Acción                |
|--------------------------|---|-------------|--------------|--------------------|-----------|------|----------------|-------------|----------------|-----------------------|
| <input type="checkbox"/> | 1 | id          | bigint(20)   |                    |           | No   | Ninguna        |             | AUTO_INCREMENT | Cambiar  Eliminar Más |
| <input type="checkbox"/> | 2 | descripcion | varchar(255) | utf8mb4_general_ci |           | Sí   | NULL           |             |                | Cambiar  Eliminar Más |
| <input type="checkbox"/> | 3 | media       | varchar(255) | utf8mb4_general_ci |           | Sí   | NULL           |             |                | Cambiar  Eliminar Más |
| <input type="checkbox"/> | 4 | titulo      | varchar(255) | utf8mb4_general_ci |           | Sí   | NULL           |             |                | Cambiar  Eliminar Más |

- Base de datos MySQL Workbench

MySQL Workbench interface showing a query result for the 'coworking' table.

Query 1: `SELECT * FROM heroku_b6b386f98a95cb0.coworking;`

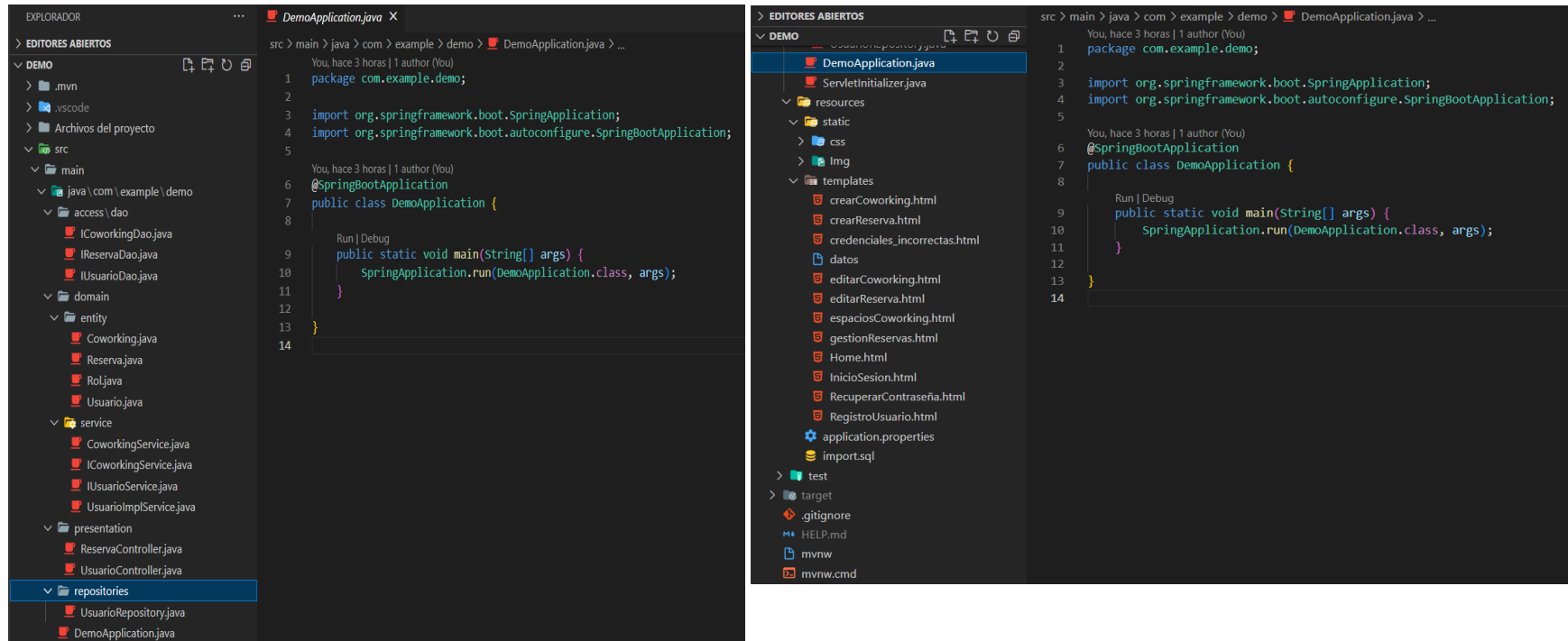
Result Grid:

| id | descripcion   | media   | titulo             |
|----|---|---|--------------------|
| 54 | Oficina con espacios de trabajo individuales y s... | <a href="https://www.ejeprime.com/files/fotos/atikko/Ati...">https://www.ejeprime.com/files/fotos/atikko/Ati...</a> | Oficina de trabajo |

Output:

| # | Time     | Action   | Message           |
|---|----------|--|-------------------|
| 1 | 22:52:39 | SELECT * FROM heroku_b6b386f98a95cb0.coworking LIMIT 0, 1000 | 1 row(s) returned |

- Pantallazo proyecto (espacio de trabajo Visual Studio Code)



## - Pantallazo pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.3</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.example</groupId>
  <artifactId>demo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>demo</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>1.8</java.version>
  </properties>
  <dependencies>
    <!-- You, here 3 horas + Proyecto final G20working -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <scope>runtime</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>
```

```
<optional>true</optional>
</dependency>
<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-tomcat</artifactId>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
</project>
```

## - Pantallazo Profile

```
1 web: java -Dserver.port=$PORT $JAVA_OPTS -Dspring.profiles.active=heroku -jar target/demo-0.0.1-SNAPSHOT.war
```

## 8. Informe de retrospectiva.

En esta cuarta etapa del desarrollo del proyecto (**SPRINT 4**) se realizaron avances de acuerdo con la respectiva elaboración del front y del back del proyecto como tal, se agregaron las correspondientes funcionalidades y características para hacer el proyecto funcional, el cual estuvo sujeto a aprobación y debidas modificaciones por parte del equipo de trabajo y contando con la respectiva supervisión de los docentes encargados. A continuación, se encuentran estipulados una serie de puntos, los cuales se trataron durante el desarrollo del SPRINT 4 para cumplir a cabalidad con la realización del proyecto en curso.

- Realizar las debidas pruebas para el correcto desarrollo del proyecto.
- Actualizaciones periódicas de los repositorios (GitHub - Trello).
- Revisiones continuas de los correspondientes repositorios del proyecto (GitHub - Trello).
- Revisiones de funcionalidades de todo el proyecto para aprobar su viabilidad.
- Despliegue del proyecto en la nube (Heroku) con conexión al GitHub del grupo.
- Se realizaron los debidos cambios en el archivo application.properties, los cuales fueron los siguientes:
  - Se agrega la url del host en heroku.
  - Se agrega el nuevo nombre de usuario de heroku.
  - Se agrega la nueva contraseña de heroku.
- Se realizan los debidos cambios en el pom.xml del proyecto, los cuales fueron los siguientes:
  - Se modifica la versión de java con base a la versión que ejecuta heroku para desplegar la app en la nube.
  - Se agrega la dependencia mysql y el `<scope>runtime</scope>`
- Se crea el correspondiente archivo Procfile para poder realizar el despliegue de la app en la nube.