



El futuro digital  
es de todos

MinTIC



# Informe - Sprint 3 del Equipo 5



Universidad de Caldas



## Tabla de participación en el desarrollo del Sprint

Integrantes (Nombre completo)	Cédula	Rol	Nivel de participación (Alto, Medio, Bajo, Retirado)
Daniel Valencia Cordero	1234097246	Administrador de Configuración	Alto
John Alexander Loaiza	75103940	Tester	Medio
Martinez Martinez Mariam Camila	1033810660	Diseñador UI	Medio
Mejia Barco Jorge Andres	9847235	Diseñador de Software	Medio
Melo Muñoz Liliana Jizzeth	1026280692	Líder de equipo	Medio

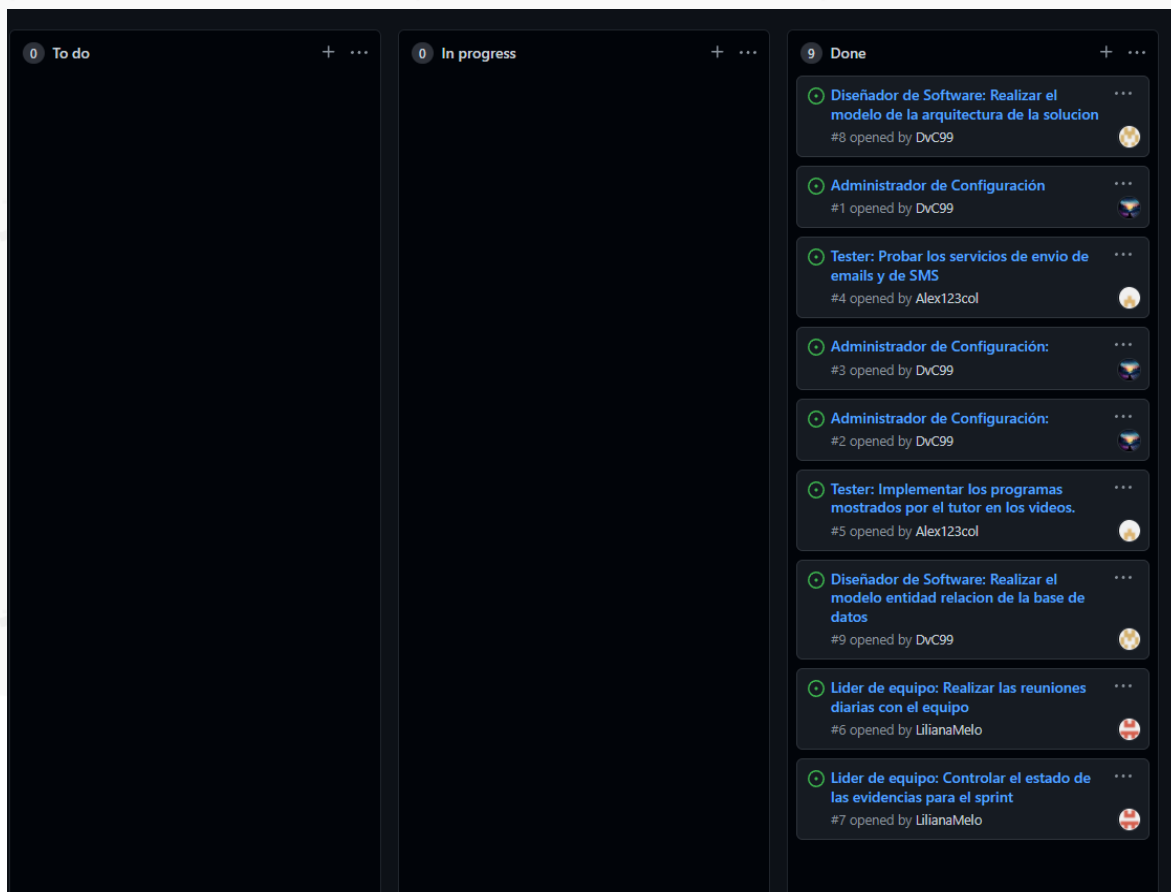


## Informe de Seguimiento del Sprint 3

**Repositorio del equipo:** <https://github.com/Developers-json>

### Kanvas del Sprint 1

Del Sprint número 1 sólo quedó pendiente por terminar la actividad de realizar el modelo de la arquitectura de la aplicación debido a que solo conocemos el Backend y todavía no sabemos cómo funciona el framework del Frontend. Esta tarea fue finalizada en este sprint debido a que empezamos con Angular.





## Kanvas del Sprint 2

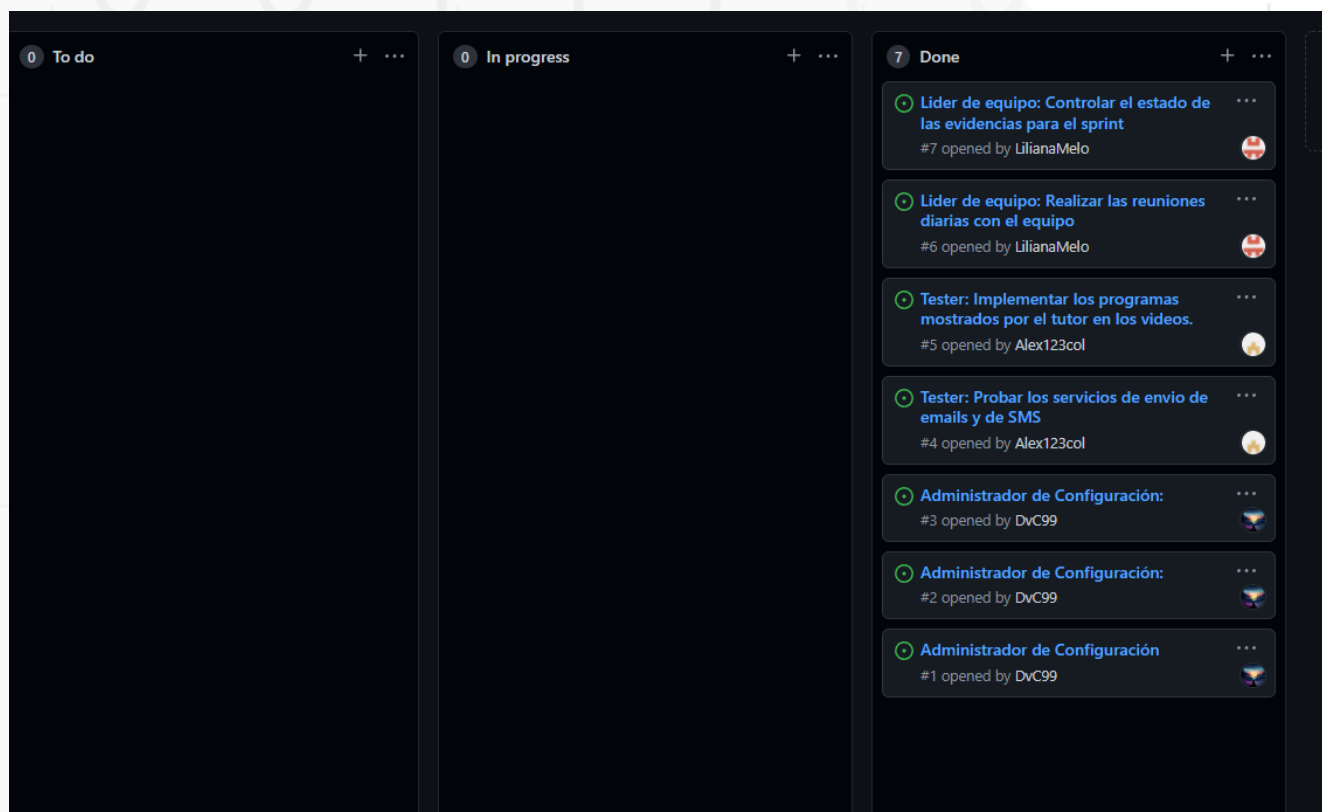
Del Sprint número 2 no quedó pendiente ninguna actividad sobre el backend y debido al evento del fin de semana no terminamos de ver por completo Angular.

Column	Count	Items
To Do	0	None
In Progress	0	None
Done	7	<ul style="list-style-type: none"><li>Administrador de Configuración #1 opened by DvC99</li><li>Lider de equipo: Realizar las reuniones diarias con el equipo #6 opened by LilianaMelo</li><li>Administrador de Configuración: #3 opened by DvC99</li><li>Lider de equipo: Controlar el estado de las evidencias para el sprint #7 opened by LilianaMelo</li><li>Tester: Probar los servicios de envio de emails y de SMS #4 opened by Alex123col</li><li>Administrador de Configuración: #2 opened by DvC99</li><li>Tester: Implementar los programas mostrados por el tutor en los videos. #5 opened by Alex123col</li></ul>



## Kanvas del Sprint 3

Del Sprint número 3 tenemos que implementar los servicios de autenticación, login, token y de seguridad en Loopback. El tester se encargó de implementar todos los videos de la semana por lo cual no quedó nada pendiente este sprint.





## Lunes:

Este día no nos reunimos, pero el tester quedó en adelantar los videos de la semana para que en la próxima reunión pudiera explicarnos y solucionar cualquier problema que tengamos.

## Martes:

El martes nos reunimos para mirar como íbamos con el desarrollo de los videos. Algunos tuvieron problemas al importar una librería en una versión incompatible con loopback. Se encontró la documentación de la librería y encontramos que la versión anterior a la más reciente era la correcta.

```
1  import {Model, model, property} from '@loopback/repository';
2
3  @model()
4  export class Credenciales extends Model {
5    @property({
6      type: 'string',
7      required: true,
8    })
9    Usuario: string;
10
11    @property({
12      type: 'string',
13      required: true,
14    })
15    Clave: string;
16
17    constructor(data?: Partial<Credenciales>) {
18      super(data);
19    }
20  }
21
22  export interface CredencialesRelations {
23    // describe navigational properties here
24  }
25
26  export type CredencialesWithRelations = Credenciales & CredencialesRelations;
```



## Miércoles:

El miércoles el tester nos informa que ya ha terminado los videos de la semana y quedamos en que nos vamos a reunir el jueves para mirar cómo se implantaba el contenido del video.

```
GenerarClave() {
  var maxLength = 18;
  var minLength = 12;
  var password = "";

  var randomLength = Math.floor(Math.random() * (maxLength - minLength)) + minLength;
  while (!isStrongEnough(password)) {
    password = generatePassword(randomLength, false, /[\w\d\?\-\-]/);
  }
  return password;
}

EncriptarClave(clave: string) {
  return CryptoJS.MD5(clave).toString();
}

GenerarCodigo() {
  var maxLength = 10;
  var minLength = 6;
  var password = "";

  var randomLength = Math.floor(Math.random() * (maxLength - minLength)) + minLength;
  password = generatePassword(randomLength, false, /[\w\d\?\-\-]/);
  return password
}

GenerarTokenJWT(persona: Persona) {
  const token = jwt.sign({
    data: {
      id: persona.id,
      correo: persona.Correo,
      telefono: persona.Celular
    }
  },
  keys.claveJWT)
  return token
}

ValidarTokenJWT(token: string) {
  try {
    const datos = jwt.verify(token, keys.claveJWT)
    return datos;
  } catch (error) {
    return false
  }
}
```



## Jueves:

Como el día anterior terminamos con los videos de la semana, solo quedaba hacer el reto que consiste en hacer la parte de recuperar y cambiar la contraseña de un usuario registrado. El día nos alcanzó para terminarlo y lo hicimos de la siguiente manera.

Creamos los modelos necesarios que son los siguientes:

### 1. Modelo cambio clave

```
import {Model, model, property} from '@loopback/repository';

@model()
export class CambioClabe extends Model {
  @property({
    type: 'string',
    id: true,
    generated: true,
  })
  id?: string;

  @property({
    type: 'string',
    required: true,
  })
  correo: string;

  constructor(data?: Partial<CambioClabe>) {
    super(data);
  }

  export interface CambioClabeRelations {
    // describe navigational properties here
  }

  export type CambioClabeWithRelations = CambioClabe & CambioClabeRelations;
```





## 2. Modelo recuperar clave

```
import {Model, model, property} from '@loopback/repository';

@model()
export class RecuperarClave extends Model {
  @property({
    type: 'string',
    id: true,
    generated: true,
  })
  id?: string;

  @property({
    type: 'string',
    required: true,
  })
  correo: string;

  @property({
    type: 'string',
    required: true,
  })
  password: string;

  @property({
    type: 'string',
    required: true,
  })
  codigo: string;

  constructor(data?: Partial<RecuperarClave>) {
    super(data);
  }
}

export interface RecuperarClaveRelations {
  // describe navigational properties here
}

export type RecuperarClaveWithRelations = RecuperarClave & RecuperarClaveRelations;
```



Creamos el controlador de persona los siguientes endpoints:

```
@post('/cambio-clave')
@response(200, {
  description: "Se genera un codigo para validar el cambio de contraseña"
})
async ValidarCambioClave(
  @requestBody() CambioClave: CambioClave
): Promise<Boolean> {
  const usuario = await this.personaRepository.findOne({
    where: {
      Correo: CambioClave.correo
    }
  })
  if (usuario) {
    const codigo = this.servicioAutenticacion.GenerarCodigo()
    usuario.Codigo = codigo
    await this.personaRepository.updateById(usuario.id, usuario)
    //Se envia la clave al celular del usuario
    const destino = usuario.Celular
    const contenido = `Hola ${usuario.Nombre} + " " + usuario.Apellido}, te enviamos el codigo de verificacion para el cambio de contraseña: ${usuario.Codigo}`
    await fetch(`${keys.urlServicioNotificaciones}/sms?mensaje=${contenido}&telefono=${destino}`)
      .then((data: any) => {
        console.log(data)
      })
    return true
  }
  return false
}
```

```
@post('/recuperar-clave')
@response(200, {
  description: "Recuperar clave del usuario"
})
async RecuperarClave(
  @requestBody() RecuperClave: RecuperarClave
): Promise<Boolean> {
  const usuario = await this.personaRepository.findOne({
    where: {
      Correo: RecuperClave.correo
    }
  })
  if (usuario) {
    if (usuario.Codigo == RecuperClave.codigo) {
      const claveEncriptada = this.servicioAutenticacion.EncriptarClave(RecuperClave.password)
      usuario.Clave = claveEncriptada
      await this.personaRepository.updateById(usuario.id, usuario)
      //Se notifica al usuario sobre el cambio de la cuenta por correo
      const destino = usuario.Celular
      const asunto = 'Notificación de Cambio de contraseña en la plataforma'
      const contenido = `Hola ${usuario.Nombre} + " " + usuario.Apellido}, te notificamos que se ha cambiado la contraseña de tu cuenta con el nombre de usuario: ${usuario.Codigo}`
      await fetch(`${keys.urlServicioNotificaciones}/mailSMTP?email=${destino}&asunto=${asunto}&mensaje=${contenido}`)
        .then((data: any) => {
          console.log(data)
        })
      return true
    } else {
      return false
    }
  }
  return false
}
```



## Viernes:

El viernes nos reunimos para estructurar el sprint 3, corregir ortografía y otros detalles, también hablamos e hicimos una lluvia de ideas para tener más claro cómo va a quedar el mockup de la página web, además que sea funcional y agradable a la vista de los usuarios de esta. El diseñador UI va a implementar el diseño en el transcurso de la semana. Los demás vamos a ver los siguientes videos.

