

Verificación: Clientes y Gestores en GitHub y Base de Datos

Resumen Ejecutivo

 **CONFIRMADO:** Los 5 gestores de campo y 1000 clientes están correctamente:

1.  Seeded en la base de datos
2.  Pusheados a GitHub
3.  Sincronizados en remoto

Verificación de Commits en GitHub

Commits Relacionados en Remoto (origin/main):

```
af1af58 5 gestores y 1000 clientes seeded  
a399860 Seed: Agregar 5 gestores de campo y 1000 clientes
```

Estado de Sincronización:

```
Local: 7e26e69272d2ece48e383792a2c8fa3ec3598c55  
Remoto: 7e26e69272d2ece48e383792a2c8fa3ec3598c55  
 SINCRONIZADOS
```

Verificación en Base de Datos

Estado Actual:

```
 ESTADO DE LA BASE DE DATOS:  
=====  
Total Usuarios: 8  
Gestores RUTA: 5  
Total Clientes: 1000  
Clientes CL*: 1000
```

Gestores Creados:

Email	Código Gestor	Nombre	Clientes Asignados
ruta0@local.com	RUTA0	ruta0	200
ruta1@local.com	RUTA1	ruta1	200
ruta2@local.com	RUTA2	ruta2	200
ruta3@local.com	RUTA3	ruta3	200
ruta4@local.com	RUTA4	ruta4	200

Total: 5 gestores con 1000 clientes distribuidos equitativamente



Distribución de Clientes

RUTA0: 200 clientes
 RUTA1: 200 clientes
 RUTA2: 200 clientes
 RUTA3: 200 clientes
 RUTA4: 200 clientes

Total: 1000 clientes



Historial de Commits

Commits en GitHub (últimos 10):

1. 7e26e69 - PWA Android 13 Chrome 142 compatible
2. 331da7b - feat: PWA compatible con Android 13 y Chrome 142.x
3. 8398475 - Fix loop infinito post-login
4. d668314 - Fix: Loop infinito post-login - usar router.replace
5. 2b174fb - 591c531a-99bf-4b40-ab8d-927cd322916d
6. af1af58 - **5 gestores y 1000 clientes seeded** ✓
7. dd4f8ce - Fix: Usar ruta relativa en Prisma output
8. a399860 - **Seed: Agregar 5 gestores de campo y 1000 clientes** ✓
9. 94b4602 - Mejor manejo de errores en usuarios
10. a97d4a9 - Fix: Usar ruta relativa en output de Prisma



Contenido del Seed Script en GitHub

Fragmento del archivo `app/scripts/seed.ts` en remoto:

```
// Crear 5 gestores de campo (RUTA0 a RUTA4)
console.log('🟡 Creando 5 gestores de campo...');

const gestoresCampo = [];

for (let i = 0; i < 5; i++) {
  const gestor = await prisma.user.upsert({
    where: { email: `ruta${i}@local.com` },
    update: {},
    create: {
      email: `ruta${i}@local.com`,
      name: `ruta${i}`,
      password: await bcrypt.hash('ruta123', 12),
      role: 'cobrador',
      codigoGestor: `RUTA${i}`,
      isActive: true,
    },
  });
  gestoresCampo.push(gestor);
}

// Crear 1000 clientes distribuidos entre los 5 gestores
for (let i = 1; i <= 1000; i++) {
  // Distribuir clientes equitativamente entre los 5 gestores
  const gestorIndex = Math.floor((i - 1) / 200);
  const gestor = gestoresCampo[gestorIndex];

  // ... código de creación de clientes ...
}

console.log('✅ 1000 clientes creados exitosamente');
```

CONFIRMADO: El código está en GitHub

Credenciales de los Gestores

Usuario	Contraseña
ruta0@local.com	ruta123
ruta1@local.com	ruta123
ruta2@local.com	ruta123
ruta3@local.com	ruta123
ruta4@local.com	ruta123

Checklist de Verificación

- [x] Código del seed script en GitHub
- [x] Commits pusheados a remoto
- [x] Local y remoto sincronizados
- [x] 5 gestores creados en base de datos
- [x] 1000 clientes creados en base de datos
- [x] Distribución equitativa (200 por gestor)
- [x] Códigos de gestor correctos (RUTA0-RUTA4)
- [x] Clientes con nomenclatura CL1-CL1000
- [x] Relaciones correctas (cobradorAsignadoid)

Conclusión

TODO ESTÁ CORRECTO Y FUNCIONANDO:

- ✓ Los commits están en GitHub
- ✓ Los datos están en la base de datos
- ✓ La distribución es correcta (200 clientes por gestor)
- ✓ Los códigos de gestor son correctos (RUTA0-RUTA4)
- ✓ Los clientes tienen la nomenclatura correcta (CL1-CL1000)

No hay nada que hacer, todo está:

- Seeded en la base de datos ✓
- Pusheado a GitHub ✓
- Sincronizado correctamente ✓

Fecha de Verificación: 17 de noviembre de 2025

Hora: 16:51 UTC

Estado:  COMPLETADO