



LABORATORIO 2

Sergio Espinosa
Gonzalo Ibáñez
Cristóbal Olivares

TABLA CONTENIDO

1. AVANCE
- FUNCIONALIDADES

2. DIAGRAMA
- ARQUITECTURA

3. DEMOSTRACIONES

4. -

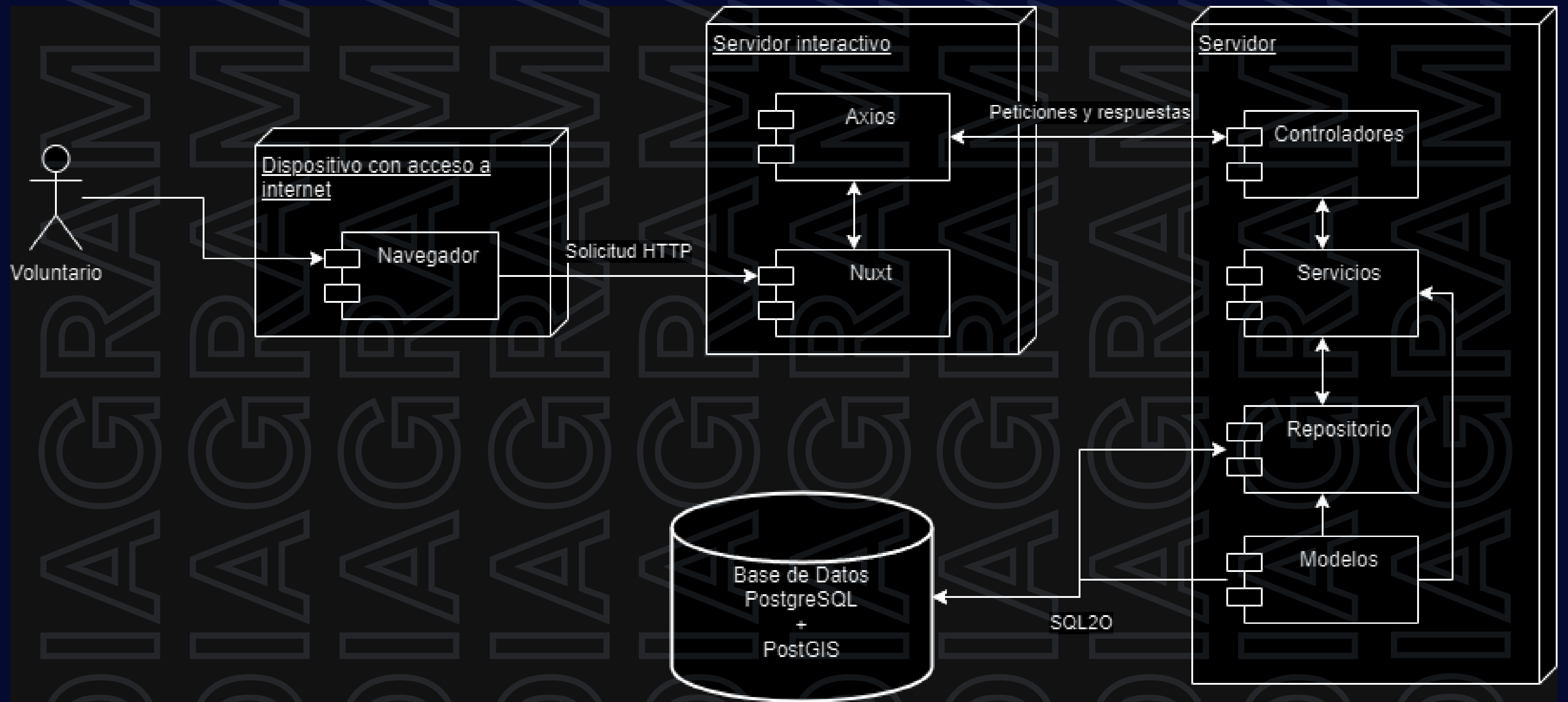
TABLA CONTENIDO

5. = CONTRIBUCIONES

6. = CONCLUSIONES

Funcionalidades - Progreso:

- 1) Se debe guardar la ubicación de la emergencia como punto en la BD (13)
- 100%
- 2) Se debe guardar la ubicación de tarea como punto en la BD (15) - 100%
- 3) Se deben guardar las regiones de Chile en la BD como poli-polígono (16)
- 100%
- 4) Se deben obtener la lista de emergencias que pertenecen a la misma
región que el voluntario actual (25) - 100%



FRAGMENTO DE CÓDIGO

```
@PostMapping
public EmergencyDto createEmergency(@RequestBody EmergencyDto dto) {
    Emergency emergency = new Emergency();
    emergency.setId_emergency(dto.getId_emergency());
    emergency.setDescription(dto.getDescription());
    emergency.setDate(dto.getDate());
    emergency.setActive(dto.getActive());
    emergency.setId_institution(dto.getId_institution());

    WKTRReader reader = new WKTRReader();
    try {
        Point location = (Point) reader.read(dto.getLocation());
        emergency.setLocation(location);
    } catch (ParseException e) {
        // Manejar la excepción
    }

    Emergency savedEmergency = service.create(emergency);
    return convertToDto(savedEmergency);
}
```

6 usages

```
private EmergencyDto convertToDto(Emergency emergency) {
    EmergencyDto dto = new EmergencyDto();
    dto.setId_emergency(emergency.getId_emergency());
    dto.setDescription(emergency.getDescription());
    dto.setDate(emergency.getDate());
    dto.setActive(emergency.getActive());
    dto.setId_institution(emergency.getId_institution());
    dto.setLocation(emergency.getLocation().toText());
    System.out.println("Exit");
    return dto;
}
```

FRAGMENTO DE CÓDIGO

The screenshot shows a PostgreSQL IDE interface. On the left, the 'Object Explorer' pane displays a tree structure of database objects. The 'demo' database is expanded, showing 'Schemas (1)' with the 'public' schema selected. The 'public' schema contains various objects, including 'Tables (14)', with the 'emergency' table highlighted. The main window displays the 'Query' tab with the following SQL query:

```
1 SELECT * FROM public.emergency
2 ORDER BY id_emergency ASC
```

Below the query, the 'Data Output' tab shows the results of the query. The results are displayed in a table with the following columns: 'id', 'description', 'date', 'active', 'id_institution', and 'location'. The table contains 7 rows of data.

		description text	date timestamp without time zone	active boolean	id_institution bigint	location geometry
1	1	Incendio en el centro de la ciudad	2023-10-18 00:24:30	true	1	0101000020E6100000963E74417DA951C0FC1D8A027DBA40C0
2	2	Inundación en la zona norte	2023-10-17 22:00:00	true	2	0101000020E6100000A835CD3B4E9951C0F437A11001A737C0
3	3	Accidente de tráfico en la autopista	2023-10-17 18:45:00	false	3	0101000020E6100000E8BCC62E51E351C03A3B191C258340C0
4	4	Deslizamiento de tierra en el parque	2023-10-17 16:30:00	false	4	0101000020E610000027A5A0DB4BE851C0F853E3A59B8440C0
5	5	Persona perdida en el bosque	2023-10-17 14:15:00	true	5	0101000020E610000056B77A4E7A4752C0F7C77BD5CA5C42...
6	6	Rescate de gato en árbol alto	2023-10-17 12:00:00	false	6	0101000020E61000000612143FC6AC51C00FD6FF39CCCB40C0
7	7	Derrame de sustancias químicas en fábrica	2023-10-17 09:45:00	true	7	0101000020E61000006E5166834C2652C0423EE8D9AC5E43C0

FRAGMENTO DE CÓDIGO

The screenshot displays a PostgreSQL client interface. On the left, the 'Object Explorer' shows a tree of database objects, with 'region' selected under 'Tables (14)'. The main window shows a query executed in the 'public.region/demo/postgres@PostgreSQL 15' database. The query is:

```
1 SELECT * FROM public.region
2 ORDER BY id_region ASC
```

The 'Data Output' tab shows the results of the query, which are 15 rows. The first row is highlighted. The 'Geometry Viewer' tab shows a map of Chile with the regions outlined in blue. The 'Data Output' tab is also visible, showing the same data.

	id_region [PK] bigint	name text	geom geometry
1	1	Tarapacá	0106000020E610000001000000010300000001000000E20300003080978A3B1F51C0FF9165C389F334C0461DDFE4BC20
2	2	Antofagasta	0106000020E6100000020000000103000000010000002B060000E3231708F62251C0F92375E1924C39C096E647B4CC23
3	3	Atacama	0106000020E61000000200000001030000000100000074060000E3231708F62251C0F92375E1924C39C0313D3520F7225
4	4	Coquimbo	0106000020E610000005000000010300000001000000B20500009C5D372BF7C51C091854AD597803DC080423953BB70
5	5	Valparaíso	0106000020E610000001000000010300000001000000AF03000084853E4599E251C007AB3550C51640C0F23184885CE2
6	6	Región del Libertador Gral. Bernardo O'Higgins	0106000020E6100000010000000103000000010000008C0300002202DB94F48151C04DE65A3EEF2441C0746F4DE60582
7	7	Región del Maule	0106000020E6100000010000000103000000010000007D0400006707DBA1D2FF51C0DA73949F765B41C0CBAD8706A7F
8	8	Región del Biobío	0106000020E610000004000000010300000001000000D40500008444017EC73152C092E19241BB0142C06B89D43E2931
9	9	Región de la Araucanía	0106000020E6100000010000000103000000010000002C040000E093A5DBA63152C0403BBD4C30CC42C01B119B95903
10	10	Región de Los Lagos	0106000020E610000043000000010300000001000000980900006D37BF3239EF51C002B7C9F38E0846C0F749F08032F0
11	11	Región Aysen	0106000020E6100000E60100000103000000680000002A1B00009C837C1CD5B752C0F33F38146E1647C024A4BBDE54B
12	12	Región de Magallanes	0106000020E61000003704000001030000007D000000611900001BE88E101A2852C0361B240F022D4AC0ED37E1440928
13	13	Región Metropolitana de Santiago	0106000020E6100000010000000103000000010000007303000004F9A55EFE8551C05F1B6E83958640C0C73E5F062A85
14	14	Región de Los Ríos	0106000020E610000008000000010300000001000000F4030000BD80DDA576F51C0FFC47B2C67C843C055E23754CCE

FRAGMENTO DE CÓDIGO

```
@PostMapping("/register")
public ResponseEntity<VoluntaryDto> register(@RequestBody VoluntaryDto dto) {
    Voluntary voluntary = new Voluntary();
    voluntary.setRut(dto.getRut());
    voluntary.setName(dto.getName());
    voluntary.setLastnames(dto.getLastnames());
    voluntary.setEmail(dto.getEmail());
    voluntary.setPassword(dto.getPassword());
    voluntary.setPhone(dto.getPhone());
    voluntary.setAvaible(dto.getAvaible());

    WKTRReader reader = new WKTRReader();
    try {
        Point location = (Point) reader.read(dto.getLocation());
        voluntary.setLocation(location);
    } catch (ParseException e) {
        // Manejar la excepción
    }

    Voluntary registeredVoluntary = service.register(voluntary);
    return new ResponseEntity<>(convertToDto(registeredVoluntary), HttpStatus.CREATED);
}
```

6 usages Ch3chS

```
private VoluntaryDto convertToDto(Voluntary voluntary) {
    VoluntaryDto dto = new VoluntaryDto();
    dto.setRut(voluntary.getRut());
    dto.setName(voluntary.getName());
    dto.setLastnames(voluntary.getLastnames());
    dto.setEmail(voluntary.getEmail());
    dto.setPassword(voluntary.getPassword());
    dto.setPhone(voluntary.getPhone());
    dto.setAvaible(voluntary.getAvaible());
    dto.setLocation(voluntary.getLocation().toText());
    return dto;
}
```

Data Output Messages Notifications								
	rut [PK] character v	name character var	lastnames character varying (60)	email character varying (255)	password character var	phone character varyin	avaible boolean	location geometry
1	12345678-9	Juan	Perez Gonzalez	juan.perez@mail.com	password1	912345678	true	0101000020E6100000963E74417DA951C0FC1D8A027DBA40...
2	23456789-0	Maria	Rodriguez Lopez	maria.rodriguez@mail.com	password2	923456789	true	0101000020E610000026016A6AD9A451C027A5A0DB4BCE40...
3	34567890-1	Pedro	Gomez Ramirez	pedro.gomez@mail.com	password3	934567890	false	0101000020E6100000A835CD3B4E9951C0F437A11001A737C0
4	45678901-2	Carlos	Sanchez Jimenez	carlos.sanchez@mail.com	password4	945678901	true	0101000020E6100000E8BCC62E51E351C03A3B191C258340C0
5	56789012-3	Ana	Martinez Fernandez	ana.martinez@mail.com	password5	956789012	false	0101000020E610000027A5A0DB4BE851C0F853E3A59B8440...

FRAGMENTO DE CÓDIGO



```
/**  
 * Columna con la ubicación del voluntario.  
 * Actualización postGIS (lab 2)  
 */  
@Column(name = "location", nullable = false, columnDefinition = "GEOMETRY(Point, 4236)")  
private Point location;
```

FRAGMENTO DE CÓDIGO

1 Ch3chS

```
@GetMapping("/{rut}")
public List<EmergencyDto> getEmergenciesByVoluntaryLocation(@PathVariable String rut) {
    List<Emergency> emergencies = service.getByVoluntaryLocation(rut);
    return emergencies.stream().map(this::convertToDto).collect(Collectors.toList());
}
```

1 usage 1 Ch3chS

```
public List<Emergency> getByVoluntaryLocation(String rut) {
    return repo.findByVoluntaryLocation(rut);
}
```

1 usage 1 Ch3chS

```
@Query(value = "SELECT emergency.* FROM voluntary JOIN region ON ST_Within(voluntary.location, region.geom) " +
    "JOIN emergency ON ST_Within(emergency.location, region.geom) WHERE voluntary.rut = :rut", nativeQuery = true)
List<Emergency> findByVoluntaryLocation(@Param("rut") String rut);
```



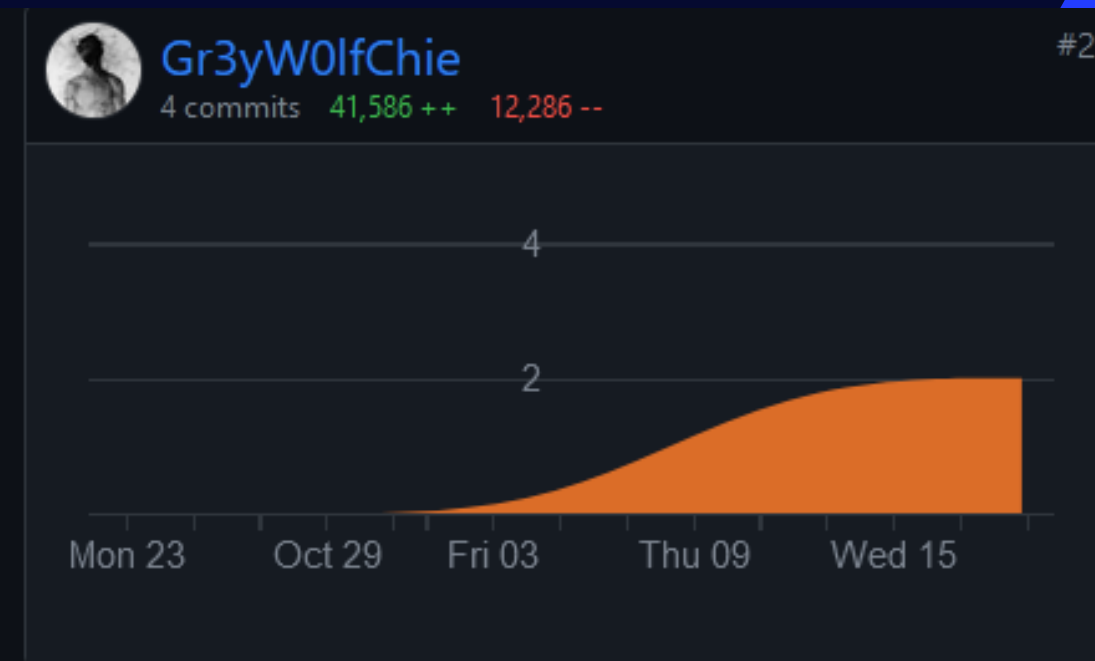
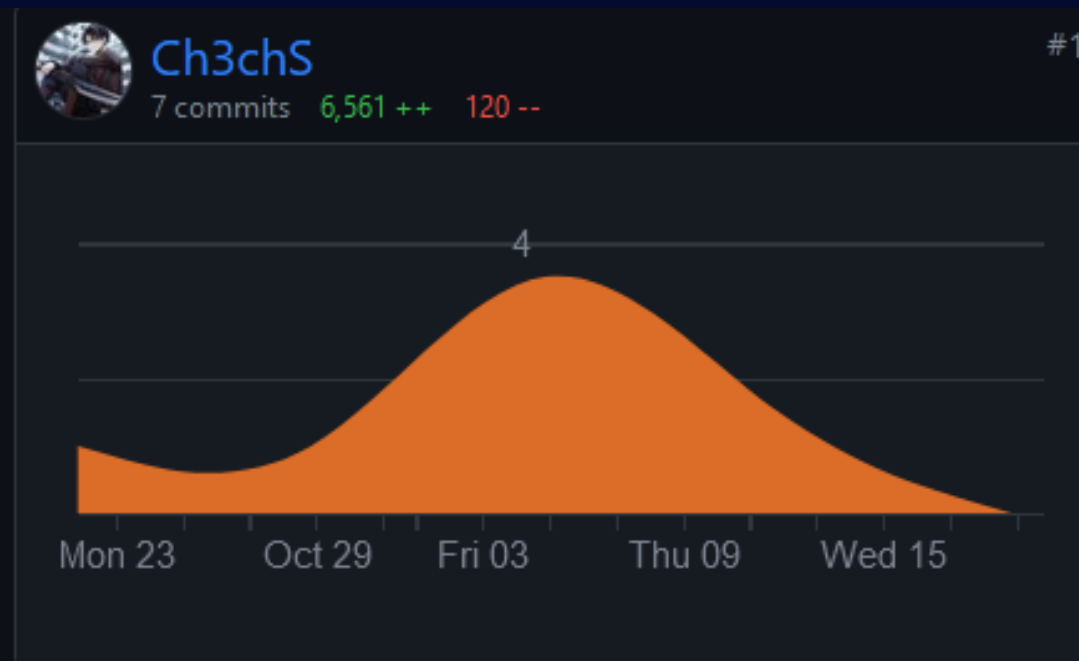
DEMOSTRACIONES



10

CONTRIBUCIONES

11



NOISISTULCONCON

Dificultades:

- Coordinación de tiempos
- Distribución de tareas
- Nuevas herramientas

Logros:

- Cumplimiento del 100% de requerimientos y funcionalidades
- Aprendizaje y desarrollo como equipo
- Aprendizaje y desarrollo a nivel personal