

DESARROLLO DE BX

1. Introducción

Resumen del Documento: Este documento describe la fase de desarrollo de Baymax, una inteligencia artificial orientada a la consulta médica y apoyo psicológico. Se presenta la estructura del código, la implementación de funcionalidades clave y los casos de prueba necesarios para validar el sistema.

Objetivo del Documento: Definir los detalles de implementación y pruebas de Baymax, documentando el proceso y resultados para confirmar que el sistema cumple con los requerimientos funcionales y no funcionales definidos previamente.

Proyecto Baymax: Estructura de Desarrollo

Organización del Repositorio

```
...

baymax-project/
|
├── frontend/
|   ├── html/
|   ├── css/
|   └── js/
|
├── backend/
|   ├── src/
|   │   ├── controllers/
|   │   ├── models/
|   │   └── services/
|   └── utils/
|
└── tests/
|
```

```
└─ database/
|   └─ migrations/
|   └─ seeds/
|
└─ docs/
    └─ design/
    └─ requirements/
...

```

Componentes Principales

1. **Frontend**: Interfaces de usuario
2. **Backend**: Lógica de negocio y procesamiento de IA
3. **Base de Datos**: Gestión de información de usuarios
4. **Servicios**: Integración con APIs externa

Consulta Médica - Baymax

Usuario: Juan Pérez

Describe tus Síntomas

Escribe aquí los síntomas que estás experimentando...

Analizar Síntomas

Resultado del Análisis

Basado en tus síntomas, te recomendamos:

- Descanso
- Mantente hidratado
- Si los síntomas persisten, consulta a un médico

Apoyo Psicológico - Baymax

😊 Feliz

😐 Neutro

😞 Triste

😰 Ansioso

Mensaje de Apoyo

Recuerda que tus emociones son válidas. Estás haciendo un gran trabajo cuidándote a ti mismo.

Baymax

Correo Electrónico

Contraseña

Iniciar Sesión

[¿No tienes cuenta? Regístrate](#)

2. Estructura del Código

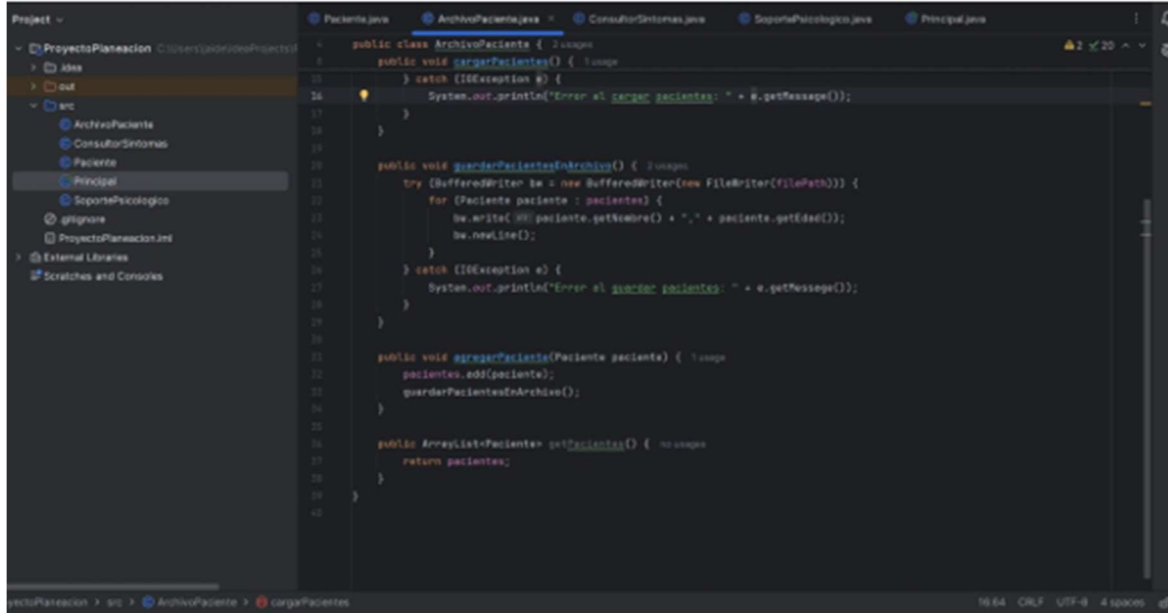
Organización del Proyecto: Baymax está desarrollado en Java utilizando el IDE IntelliJ IDEA, con una estructura organizada en directorios:

- src/: Contiene el código fuente principal.
- test/: Clases para pruebas unitarias e integradas.
- resources/: Archivos de configuración y recursos estáticos (como guías de síntomas y mensajes de apoyo).
- database/: Archivos de persistencia de datos, como pacientes.txt, manejado por la clase ArchivoPaciente.

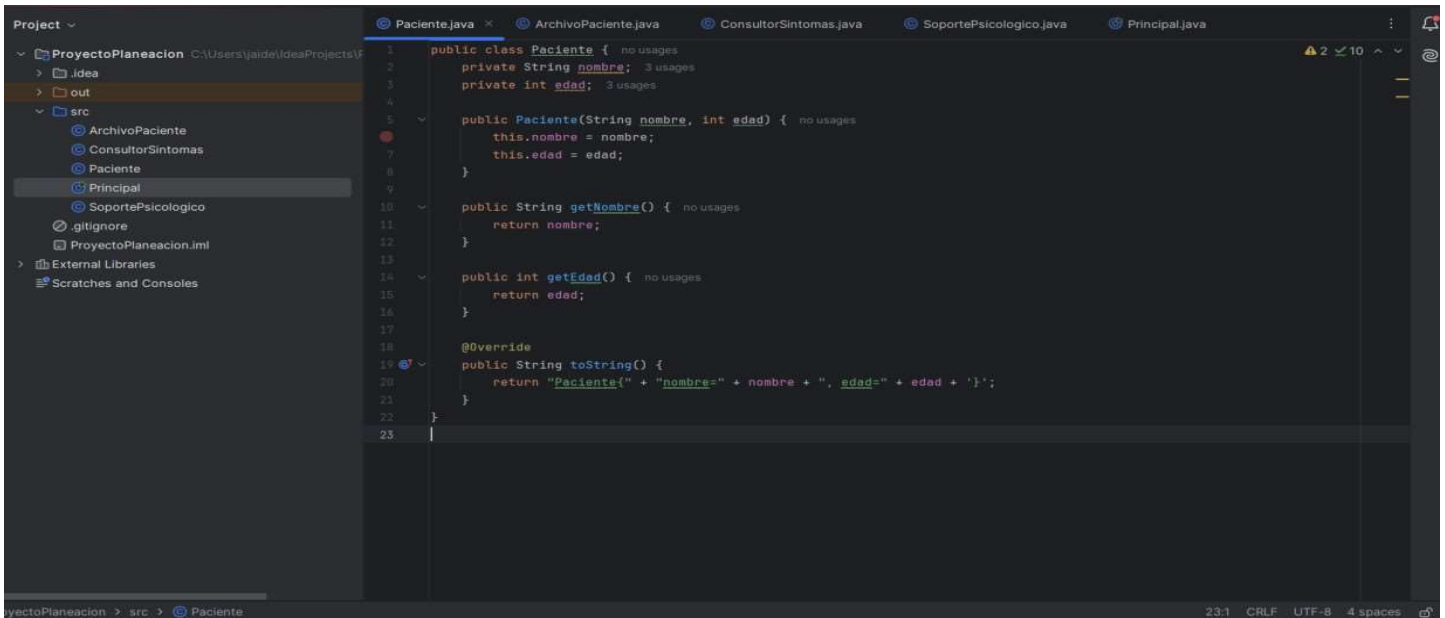
3. Estructura de Clases y Componentes

Clase Paciente

Define un paciente, con atributos básicos como nombre y edad.



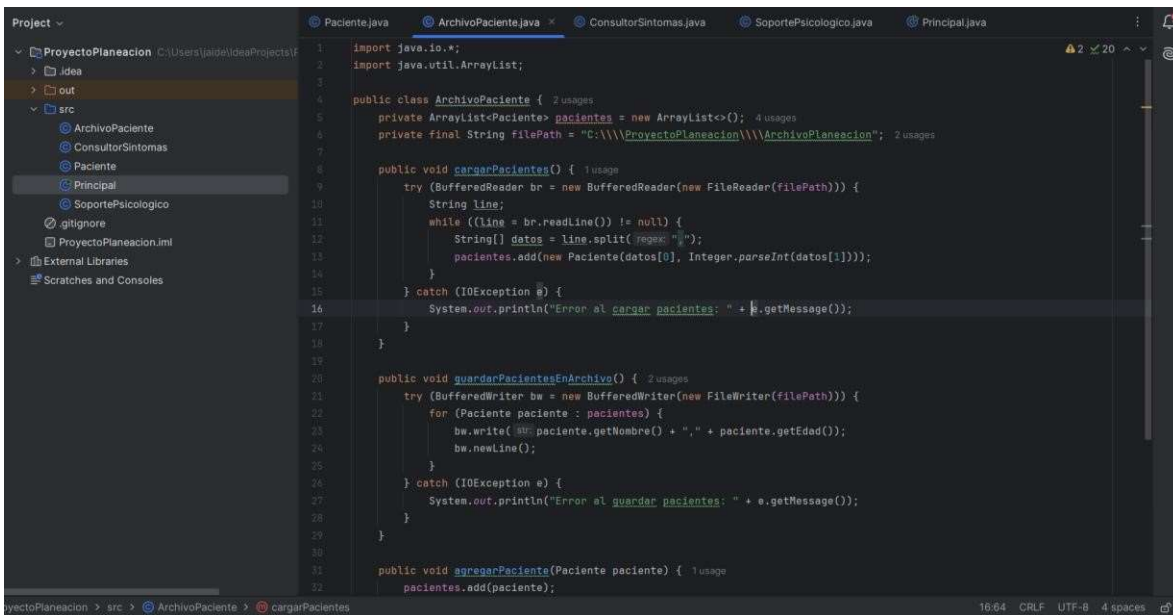
```
1 public class ArchivoPaciente {
2     public void cargarPacientes() {
3         try {
4             System.out.println("Error al cargar pacientes: " + e.getMessage());
5         } catch (IOException e) {
6             System.out.println("Error al cargar pacientes: " + e.getMessage());
7         }
8     }
9
10    public void guardarPacientesEnArchivo() {
11        try (BufferedWriter bw = new BufferedWriter(new FileWriter(filePath))) {
12            for (Paciente paciente : pacientes) {
13                bw.write(paciente.getNombre() + "," + paciente.getEdad());
14                bw.newLine();
15            }
16        } catch (IOException e) {
17            System.out.println("Error al guardar pacientes: " + e.getMessage());
18        }
19    }
20
21    public void agregarPaciente(Paciente paciente) {
22        pacientes.add(paciente);
23        guardarPacientesEnArchivo();
24    }
25
26    public ArrayList<Paciente> getPacientes() {
27        return pacientes;
28    }
29 }
```



```
1 public class Paciente {
2     private String nombre;
3     private int edad;
4
5     public Paciente(String nombre, int edad) {
6         this.nombre = nombre;
7         this.edad = edad;
8     }
9
10    public String getNombre() {
11        return nombre;
12    }
13
14    public int getEdad() {
15        return edad;
16    }
17
18    @Override
19    public String toString() {
20        return "Paciente{" + "nombre=" + nombre + ", edad=" + edad + '}';
21    }
22 }
```

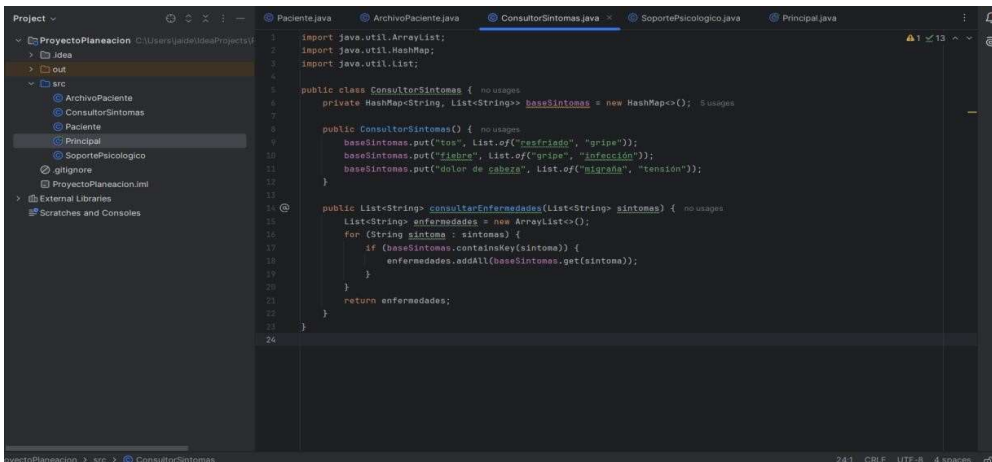
Clase ArchivoPaciente

Se encarga de la persistencia de los datos de los pacientes, cargándolos desde pacientes.txt y guardando cambios cuando se actualiza la información.



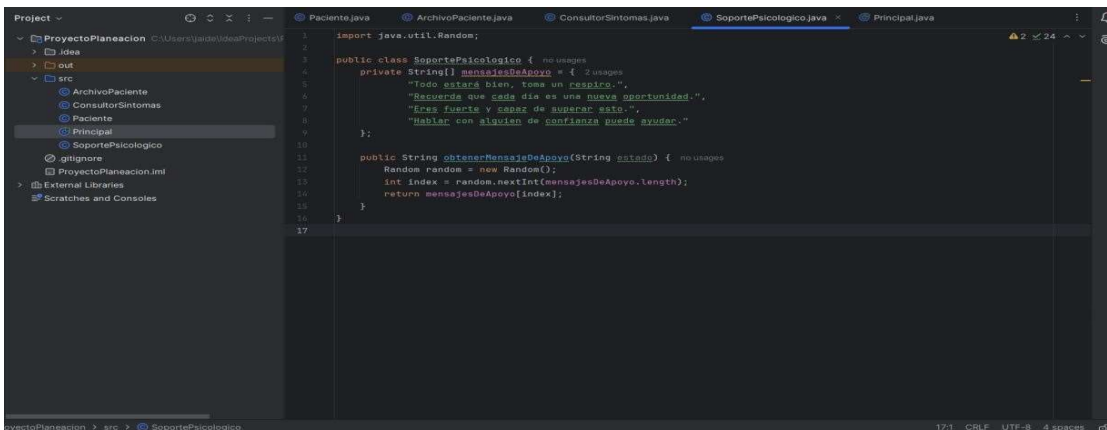
```
1 import java.io.*;
2 import java.util.ArrayList;
3
4 public class ArchivoPaciente {
5     private ArrayList<Paciente> pacientes = new ArrayList<>();
6     private final String filePath = "C:\\\\ProyectoPlaneacion\\\\ArchivoPlaneacion";
7
8     public void cargarPacientes() {
9         try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {
10             String line;
11             while ((line = br.readLine()) != null) {
12                 String[] datos = line.split(";");
13                 pacientes.add(new Paciente(datos[0], Integer.parseInt(datos[1])));
14             }
15         } catch (IOException e) {
16             System.out.println("Error al cargar pacientes: " + e.getMessage());
17         }
18     }
19
20     public void guardarPacientesEnArchivo() {
21         try (BufferedWriter bw = new BufferedWriter(new FileWriter(filePath))) {
22             for (Paciente paciente : pacientes) {
23                 bw.write(paciente.getNombre() + ";" + paciente.getEdad());
24                 bw.newLine();
25             }
26         } catch (IOException e) {
27             System.out.println("Error al guardar pacientes: " + e.getMessage());
28         }
29     }
30
31     public void agregarPaciente(Paciente paciente) {
32         pacientes.add(paciente);
33     }
34 }
```

Clase CosultorSintomas Esta clase permite ingresar síntomas y sugiere posibles enfermedades utilizando una base de datos estática.



```
1 import java.util.ArrayList;
2 import java.util.HashMap;
3 import java.util.List;
4
5 public class ConsultorSintomas {
6     private HashMap<String, List<String>> baseSintomas = new HashMap<>();
7
8     public ConsultorSintomas() {
9         baseSintomas.put("tos", List.of("resfriado", "gripe"));
10        baseSintomas.put("fiebre", List.of("gripe", "infección"));
11        baseSintomas.put("dolor de cabeza", List.of("migraña", "tensión"));
12    }
13
14    public List<String> consultarEnfermedades(List<String> sintomas) {
15        List<String> enfermedades = new ArrayList<>();
16        for (String sintoma : sintomas) {
17            if (baseSintomas.containsKey(sintoma)) {
18                enfermedades.addAll(baseSintomas.get(sintoma));
19            }
20        }
21        return enfermedades;
22    }
23 }
```

Clase SoportePsicologico Ofrece mensajes de apoyo psicológico personalizados según el estado de ánimo del usuario.



```
1 import java.util.Random;
2
3 public class SoportePsicologico {
4     private String[] mensajesDeApoyo = {
5         "Todo estará bien, toma un respiro.",
6         "Recuerda que cada día es una nueva oportunidad.",
7         "Eres fuerte y capaz de superar esto.",
8         "Hablar con alguien de confianza puede ayudar."
9     };
10
11     public String obtenerMensajeDeApoyo(String estado) {
12         Random random = new Random();
13         int index = random.nextInt(mensajesDeApoyo.length);
14         return mensajesDeApoyo[index];
15     }
16 }
```

Clase Pricipal Clase de ejecución principal del sistema Baymax.

```
1 import java.util.List;
2 import java.util.Scanner;
3
4 public class Principal {
5     private static ArchivoPaciente archivoPaciente = new ArchivoPaciente(); // 3 usages
6     private static ConsultorSintomas consultorSintomas = new ConsultorSintomas(); // 1 usage
7     private static SoportePsicologico soportePsicologico = new SoportePsicologico(); // 1 usage
8
9     public static void main(String[] args) {
10         archivoPaciente.cargarPacientes();
11         Scanner scanner = new Scanner(System.in);
12
13         int opcion;
14         do {
15             System.out.println("\n1. Agregar paciente");
16             System.out.println("2. Consultar sintomas");
17             System.out.println("3. Apoyo psicológico");
18             System.out.println("4. Salir");
19             System.out.print("Seleccione una opción: ");
20             opcion = scanner.nextInt();
21             scanner.nextLine();
22
23             switch (opcion) {
24                 case 1:
25                     System.out.print("Nombre del paciente: ");
26                     String nombre = scanner.nextLine();
27                     System.out.print("Edad del paciente: ");
28                     int edad = scanner.nextInt();
29                     archivoPaciente.agregarPaciente(new Paciente(nombre, edad));
30                     System.out.println("Paciente agregado.");
31                     break;
32             }
33         } while (opcion != 4);
34
35         archivoPaciente.guardarPacientesEnArchivo();
36         scanner.close();
37     }
38 }
```

```
33         } while (opcion != 4);
34
35         archivoPaciente.guardarPacientesEnArchivo();
36         scanner.close();
37     }
38 }
```

4. Pruebas del Sistema

```
Run Principal
C:\Users\jaide\jdk\openjdk-20.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=58614:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.3\bin" -Dfile.encoding=UTF-8
1. Agregar paciente
2. Consultar síntomas
3. Apoyo psicológico
4. Salir
Seleccione una opción: 1
Nombre del paciente: Jaider
Edad del paciente: 25
Paciente agregado.

1. Agregar paciente
2. Consultar síntomas
3. Apoyo psicológico
4. Salir
Seleccione una opción: 2
Ingrese síntomas separados por comas (ej. tos, fiebre): tos, fiebre
Posibles enfermedades: [resfriado, gripe, gripe, infección]

1. Agregar paciente
2. Consultar síntomas
3. Apoyo psicológico
4. Salir
Seleccione una opción: 3
Estado de ánimo del usuario (ej. triste): triste
Hablar con alguien de confianza puede ayudar.

1. Agregar paciente
2. Consultar síntomas
3. Apoyo psicológico
4. Salir
Seleccione una opción: 4
```

-- Modelo de Desarrollo de Base de Datos para Baymax

-- Basado en el análisis previo y considerando las necesidades de privacidad creamos una base de datos escalable y con función en cascada con el fin de que se le pueda hacer mantenimiento y llegado el caso modificar el sistema para posibles actualizaciones.

-- Tabla de Usuarios: Gestión de usuarios del sistema

```
CREATE TABLE usuarios (  
    id SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    email VARCHAR(150) UNIQUE NOT NULL,  
    fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    es_afiliado BOOLEAN DEFAULT FALSE,  
    ultima_conexion TIMESTAMP  
);
```

-- Tabla de Empresas de Salud: Registro de instituciones médicas

```
CREATE TABLE empresas_salud (  
    id SERIAL PRIMARY KEY,  
    nombre VARCHAR(200) NOT NULL,  
    url_base_datos VARCHAR(255),  
    normativas_privacidad TEXT  
);
```

-- Tabla de Usuarios Afiliados: Detalles específicos de usuarios con afiliación médica

```
CREATE TABLE usuarios_afiliados (  
    id SERIAL PRIMARY KEY,  
    usuario_id INTEGER REFERENCES usuarios(id),  
    empresa_salud_id INTEGER REFERENCES empresas_salud(id),  
    fecha_afiliacion DATE NOT NULL,  
    permiso_compartir_informacion BOOLEAN DEFAULT FALSE,  
    estado_afiliacion VARCHAR(50)  
);
```

-- Tabla de Consultas: Registro de consultas realizadas

```
CREATE TABLE consultas (  
    id SERIAL PRIMARY KEY,  
    usuario_id INTEGER REFERENCES usuarios(id),  
    fecha_consulta TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    sintomas TEXT,  
    tipo_consulta VARCHAR(50), -- informativa, detallada  
    respuesta_ia TEXT,  
    nivel_confianza NUMERIC(5,2)  
);
```

-- Tabla de Resultados Médicos: Almacenamiento de resultados para usuarios afiliados

```
CREATE TABLE resultados_medicos (  
    id SERIAL PRIMARY KEY,  
    usuario_afiliado_id INTEGER REFERENCES usuarios_afiliados(id),  
    fecha_resultado DATE NOT NULL,  
    detalles_resultado JSONB, -- Formato flexible para diferentes tipos de resultados  
    archivo_adjunto VARCHAR(255), -- Opcional: para almacenar referencias a documentos  
    categoria VARCHAR(100)  
);
```

-- Tabla de Citas Médicas: Programación y seguimiento de citas

```
CREATE TABLE citas_medicas (  
    id SERIAL PRIMARY KEY,  
    usuario_afiliado_id INTEGER REFERENCES usuarios_afiliados(id),  
    fecha_cita TIMESTAMP NOT NULL,  
    medico VARCHAR(200),  
    especialidad VARCHAR(100),  
    estado_cita VARCHAR(50) DEFAULT 'pendiente',  
    notas TEXT
```


);

-- Índices para optimización

```
CREATE INDEX idx_usuarios_email ON usuarios(email);
```

```
CREATE INDEX idx_consultas_usuario ON consultas(usuario_id);
```

```
CREATE INDEX idx_citas_usuario ON citas_medicas(usuario_afiliado_id);
```

