
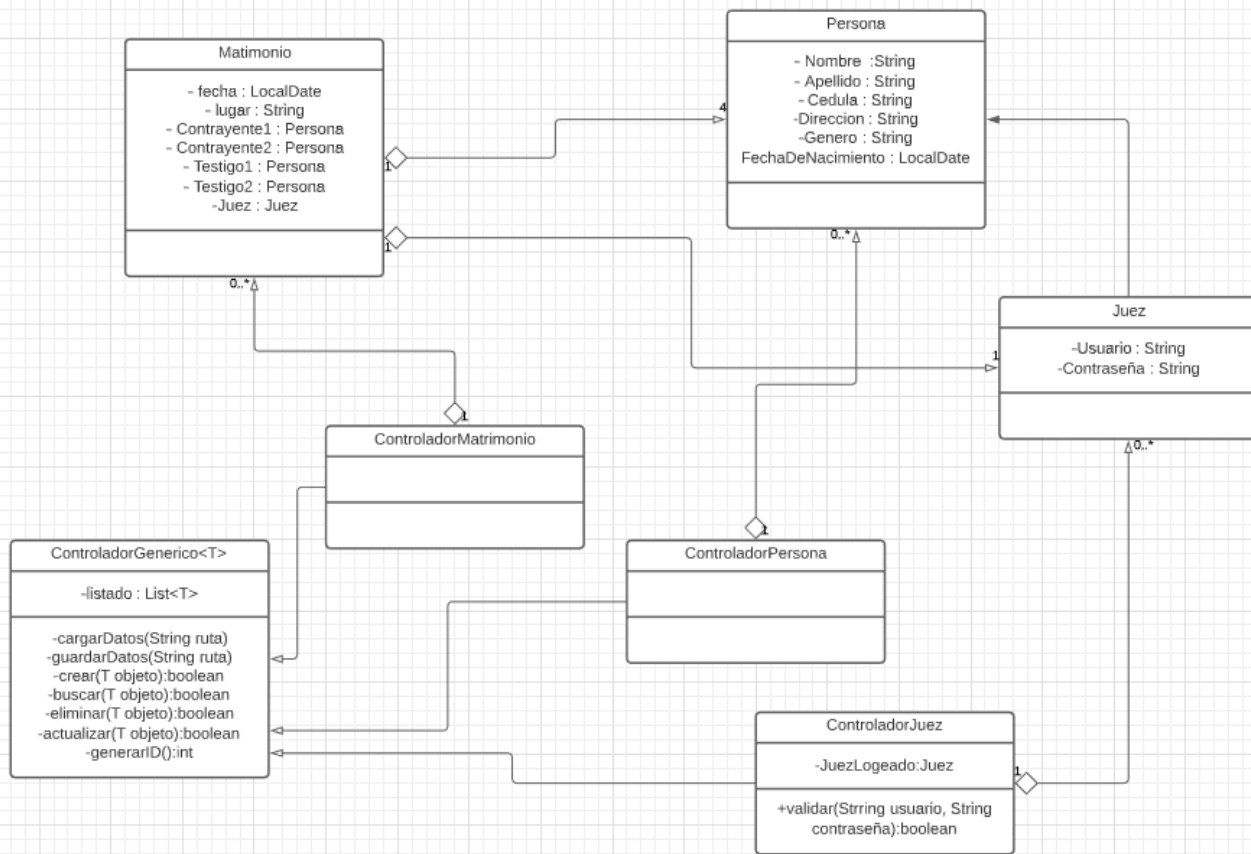
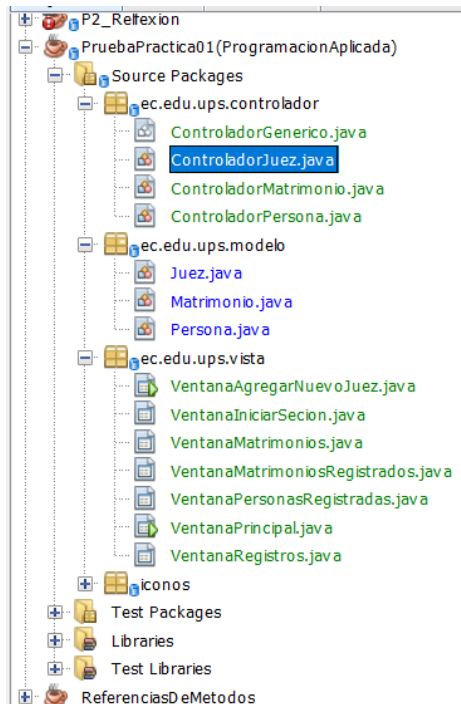
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

			PRÁCTICA DE LABORATORIO		
CARRERA: COMPUTACIÓN/INGENIERÍA DE SISTEMAS			ASIGNATURA: PROGRAMACIÓN APLICADA		
NRO. Proyecto	1.1	TÍTULO PROYECTO: Prueba Practica 1			
OBJETIVO ALCANZADO: <ul style="list-style-type: none"> Reforzar los conocimientos adquiridos en clase sobre la programación aplicada (Java 8, Programación genérica, Reflexión y Patrones de Diseño) en un contexto real. 					
ACTIVIDADES DESARROLLADAS					
1. Revisar el contenido teórico y práctico del tema <ul style="list-style-type: none"> Todos estos conceptos los hemos venido revisando y aplicando en todas las clases de Programación Aplicada, así como también en las anteriores prácticas. 					
2. Profundizar los conocimientos revisando los libros guías, los enlaces contenidos en los objetos de aprendizaje Java y la documentación disponible en fuentes académicas en línea. <ul style="list-style-type: none"> Se han revisado todos los contenidos disponibles en las fuentes académicas en línea 					
3. Deberá desarrollar un sistema informático para la gestión de matrimonios, almacenar en archivos y una interfaz gráfica. <ul style="list-style-type: none"> Se diseño el diagrama de clases para la creación del sistema 					



- Se uso el patrón MVC:



- Se crearon las clases Persona Matrimonio y Juez de las cuales Juez hereda de Persona dentro del paquete modelo

```
public class Persona implements Serializable{  
  
    private int codigo;  
    private String nombre;  
    private String apellido;  
    private String cedula;  
    private String direccion;  
    private String Genero;  
    private LocalDate fechaDeNacimiento;  
    private String estadoCivil;  
  
    public Persona() {  
    }  
}
```

```
public class Matrimonio implements Serializable  
  
    private int codigo;  
    private LocalDate fecha;  
    private String lugarDeCelebracion;  
    private Persona contrayente1;  
    private Persona contrayente2;  
    private Persona testigo1;  
    private Persona testigo2;  
    private Juez juez;  
  
    public Matrimonio() {  
    }  
}
```

```
*/  
public class Juez extends Persona implements Serializable{  
  
    private String usuario;  
    private String contraseña;  
  
    public Juez(String usuario, String contraseña) {  
        this.usuario = usuario;  
        this.contraseña = contraseña;  
    }  
}
```

- Se crearon los controladores (los cuales tendrán los métodos CRUD) los cuales implementan un patrón de diseño Singleton el cual nos garantiza que solo tendremos una instancia de dicho controlador

```
public abstract class ControladorGenerico<T> {

    private List<T> listado;

    public ControladorGenerico() {
        listado = new ArrayList<T>();
    }

    public void cargarDatos(String ruta) throws FileNotFoundException, IOException, ClassNotFoundException {
        FileInputStream archivo = new FileInputStream(ruta);
        ObjectInputStream datos = new ObjectInputStream(archivo);
        listado = (List<T>) datos.readObject();
    }

    public void guardarDatos(String ruta) throws FileNotFoundException, IOException {
        FileOutputStream archivo = new FileOutputStream(ruta);
        ObjectOutputStream datos = new ObjectOutputStream(archivo);
        datos.writeObject(listado);
    }

    public boolean crear(T objeto) {
        return listado.add(objeto);
    }

}


import ec.edu.ups.modelo.Matrimonio;

/**
 *
 * @author ariel
 */
public class ControladorMatrimonio extends ControladorGenerico<Matrimonio>{

    private static ControladorMatrimonio controladorMatrimonio;

    private ControladorMatrimonio() {
    }

    public static ControladorMatrimonio getInstancia() {
        if (controladorMatrimonio == null) {
            controladorMatrimonio = new ControladorMatrimonio();
        }
        return controladorMatrimonio;
    }
}
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

*/
public class ControladorJuez extends ControladorGenerico<Juez>{

    private static ControladorJuez controladorJuez;
    private Juez juezLogeado;

    private ControladorJuez() {
    }

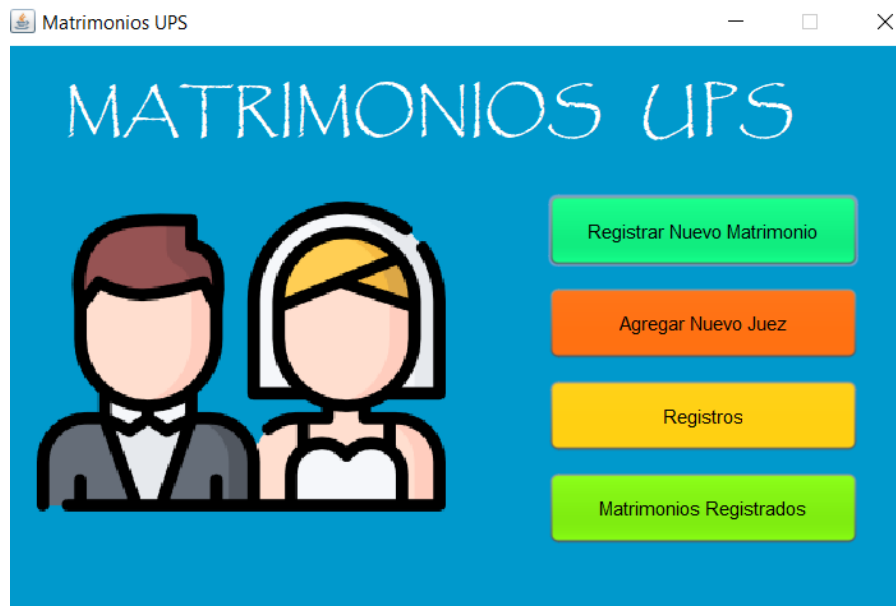
    public Juez getJuezLogeado() {
        return juezLogeado;
    }


    public static ControladorJuez getInstance() {
        if (controladorJuez == null) {
            controladorJuez = new ControladorJuez();
        }
        return controladorJuez;
    }

    public boolean validar(String usuario, String contraseña) {
        for (Juez juez : getListado()) {
            if (juez.getUsuario().equals(usuario) && juez.getContraseña().equals(contraseña)) {
                juezLogeado = juez;
                return true;
            }
        }
        return false;
    }
}

```

- Una vez hecha concluido con el modelo y el controlador se procedió con el diseño de la interfaz grafica
- Ventana principal (En esta ventana se inicializa cada uno de los controladores y ventanas además de la carga de los archivos guardados)

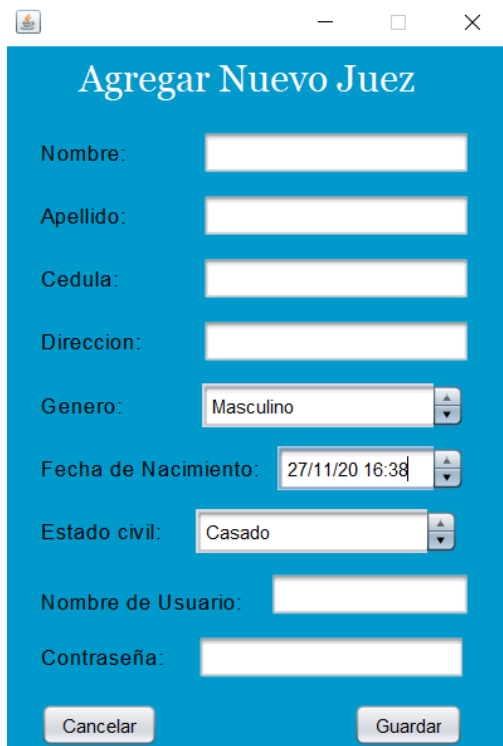



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

- Ventana de logeo a la cual se accede al dar click en Registrar Nuevo Matrimonio



- Ventana Agregar nuevo Juez



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


- Ventana Matrimonios Registrados



- Ventana Registrar Un nuevo Matrimonio

Luego de que un juez se logee de manera correcta dicho juez quedara guardado en la variable juez del controladorJueces para que en la siguiente ventana aparezca directamente dicho juez logeado

Para ello creamos un juez

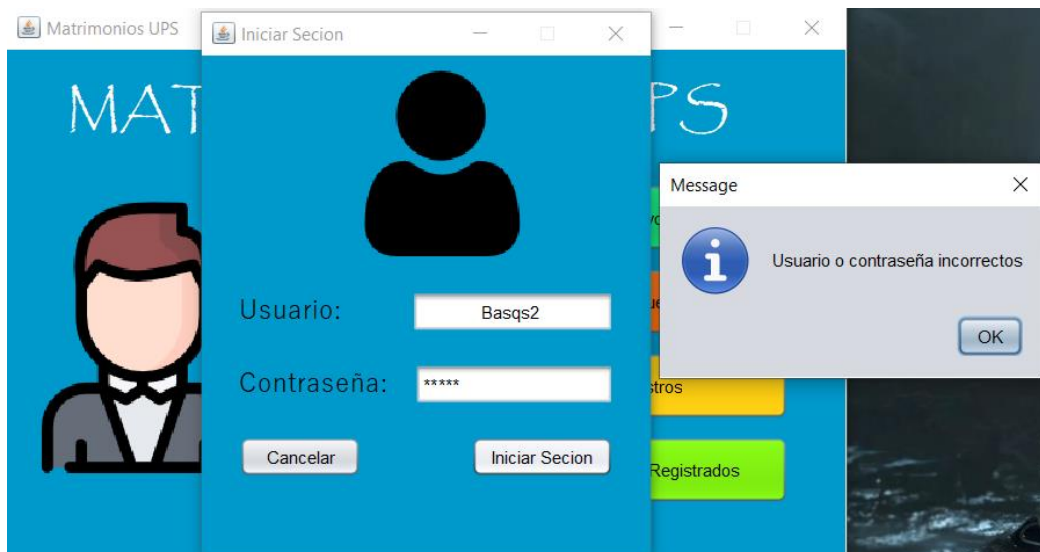



Podemos ver que aparece el juez creado en los registros



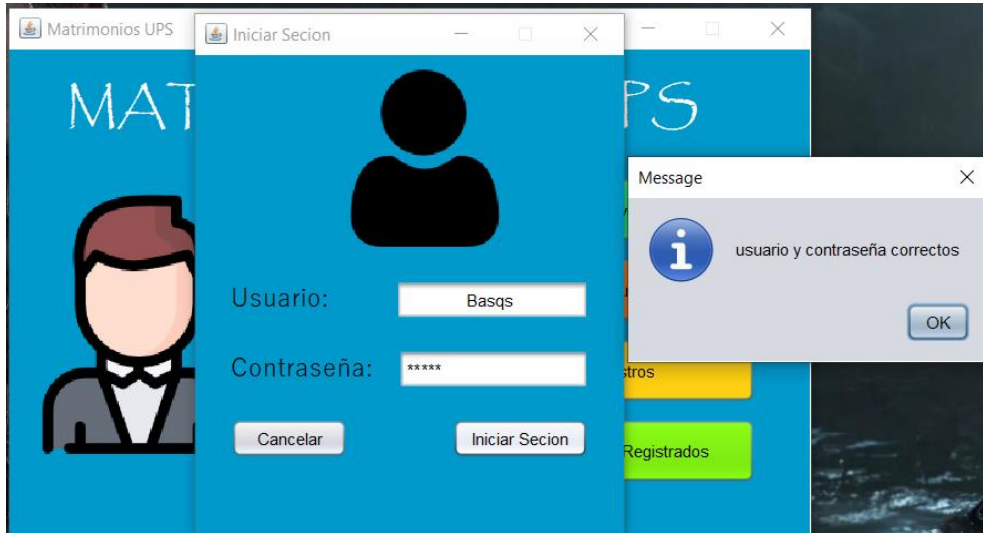
Y luego nos logeamos

Con ayuda del método validar en el controlador Jueces podemos garantizar la seguridad de nuestro programa



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Si el logeo es exitoso:

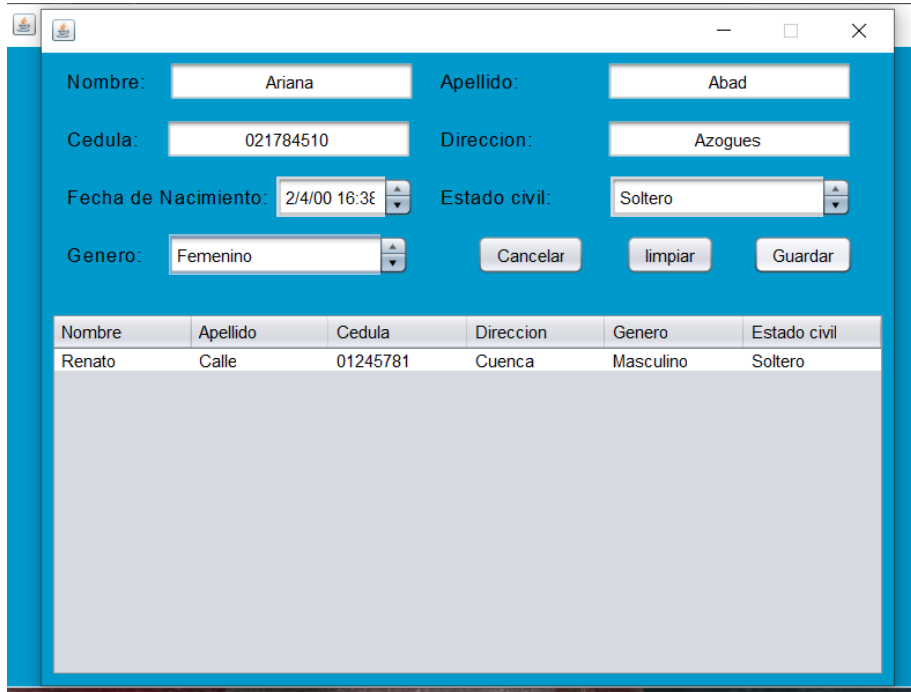


Entonces podemos entrar a la ventana para registrar un nuevo matrimonio



Desde esta ventana podemos seleccionar los contrayentes y testigos para la boda, el juez y la fecha se asignan automáticamente

Si le damos clic en el botón seleccionar a lado del JTextfile que queremos llenar nos llevara a la siguiente ventana donde podremos escoger la persona o crearla



Nombre: Ariana Apellido: Abad

Cedula: 021784510 Direccion: Azogues

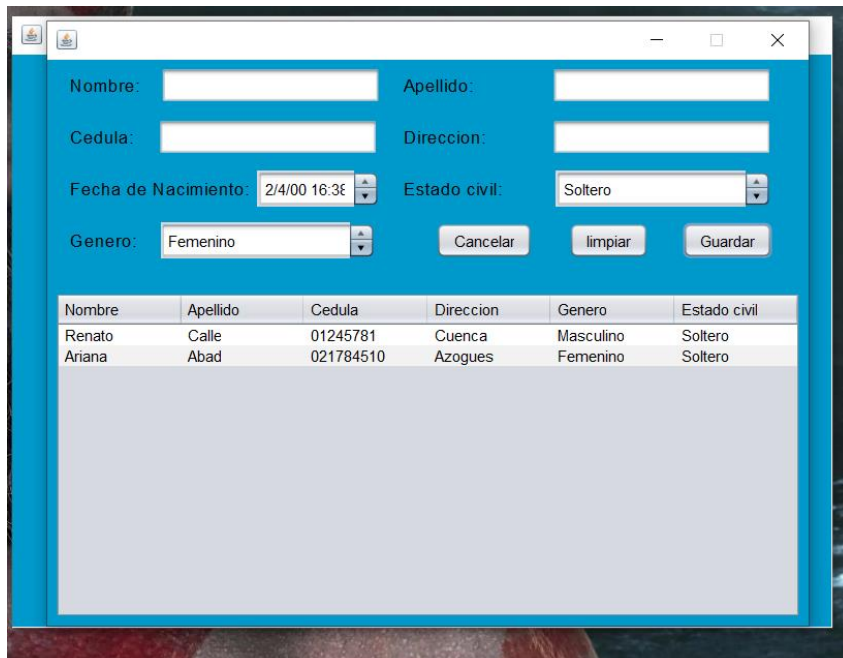
Fecha de Nacimiento: 2/4/00 16:38 Estado civil: Soltero

Genero: Femenino

Cancelar limpiar Guardar

Nombre	Apellido	Cedula	Direccion	Genero	Estado civil
Renato	Calle	01245781	Cuenca	Masculino	Soltero

Aquí podemos crear las personas que deseemos y cuando escojamos una de la tabla esta ventana se cerrara y se llenara el JText anterior:



Nombre: Apellido:


Cedula: Direccion:

Fecha de Nacimiento: 2/4/00 16:38 Estado civil: Soltero

Genero: Femenino

Cancelar limpiar Guardar

Nombre	Apellido	Cedula	Direccion	Genero	Estado civil
Renato	Calle	01245781	Cuenca	Masculino	Soltero
Ariana	Abad	021784510	Azogues	Femenino	Soltero

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Nuevo Matrimonio

Primer Contrayente:
Primer Testigo:

Cedula:
Cedula:

Segundo Contrayente:
Primer Testigo:

Cedula:
Cedula:

Lugar de Celebracion:

Juez:

Fecha:

Y de esta manera podemos seleccionar y crear a los testigos y contrayentes


Nombre: **Apellido:**

Cedula: **Direccion:**

Fecha de Nacimiento: **Estado civil:**

Genero:

Nombre	Apellido	Cedula	Direccion	Genero	Estado civil
Renato	Calle	01245781	Cuenca	Masculino	Soltero
Ariana	Abad	021784510	Azogues	Femenino	Soltero
Alvara	Vazquez	0302610076	AZigues	Masculino	Soltero
Fredy	Gonzales	1478569321	Guapan	Masculino	Soltero
Dayana	Calle	01236974	Cuenca	Masculino	Soltero

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


Luego validamos si ambos son solteros si es que los son el matrimonio se guardara exitosamente



También las personas quedan guardadas en la ventana principal (al hacer click en registros)



Nombre	Apellido	Cedula	Direccion	Genero	Estado civil
Renato	Calle	01245781	Cuenca	Masculino	Soltero
Ariana	Abad	021784510	Azogues	Femenino	Soltero
Alvara	Vazquez	0302610076	AZigues	Masculino	Soltero
Fredy	Gonzales	1478569321	Guapan	Masculino	Soltero
Dayana	Calle	01236974	Cuenca	Masculino	casado

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Y en la ventana Registros De Matrimonios aparece el matrimonio efectuado



4. Deberá generar un informe de la practica en formato PDF y en conjunto con el código se debe subir al GitHub personal.

<https://github.com/VazquezAriel/PruebaPractica01/tree/master>

RECOMENDACIONES:

- Revisar la información proporcionada por el docente previo a la práctica.
- Haber asistido a las sesiones de clase.
- **Consultar con el docente las dudas que puedan surgir al momento de realizar la prueba.**

Nombre de estudiante: Ariel Vazquez



Firma de estudiante: