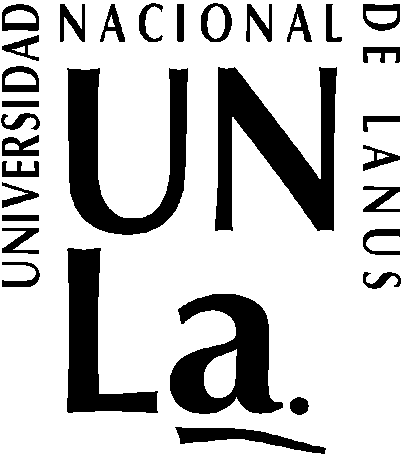
****

**Llamado a Procedimiento Remoto (RPC)**

**Docentes: Azcurra, Diego & Amaro Marcos**

**Fiorentino Kevin Emanuel**

**Violi Pablo Ezequiel**

Cátedra Desarrollo de Software en sistemas distribuidos

Licenciatura en Sistemas

2018

***Entorno de Desarrollo***

Para el proyecto se utilizaron los siguientes componentes:

Un cliente montado en una máquina virtual con Windows 7 32 bits, y funcionando en XAMP 7.2.9 con Apache.

Un servidor físico montado en Windows 10 64 bits con NetBeans 8.2, MySQL y una conexión por el driver ODBC.

Para conectar el cliente con el servidor se utilizó la versión de Thrift disponibilizada por la cátedra.

Se utilizó **PHP 5..6** para generar el servidor web. Además, se utilizó **Bootstrap 4** para los estilos

Para el servidor, se utilizo Java 8, junto con los driveres correspondientes de ODBC para conectar con la base de datos.

***Librerías Utilizadas***

*Libthrift-0.11.0* – Utilizamos esta librería JAVA para poder vincular el servidor y el cliente mediante Apache Thrift.

*Slf4j-api-1.7.25* – Se utilizó esta librería para facilitar el logging del framework.

*MySQL JDBC Driver* – se utilizó esta librería para lograr una conexión mediante ODBC

**Métodos Utilizados**

Las consultas con la base de datos MySQL se realizan por medio del conector ODBC 5.3. Se invocan desde la capa DAO ubicada en el servidor. En esta capa se creó un archivo llamado DataSource.java donde se crearon las funciones para iniciar una conexión por ODBC, y un archivo PoliDAO.java, donde se declararon las llamadas a las funciones y procedimientos de la base de datos

**Procedimientos:**

Los Procedimientos utilizados mediante thrift en el TP son:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Funcion Thrift** | **Parámetros de Entrada** | **Salida** | **Detalle** | **Procedimiento MySQL Invocado** |
| traerCanchaPorId | idCancha | Estructura tipo Cancha | Busca en la base de datos una cancha por ID, y devuelve una estructura del tipo Cancha | TRAER\_CANCHA\_POR\_IDCANCHA |
| traerCanchasPorDeporte | Deporte | Lista de Estructuras tipo Cancha | Busca en la base de datos las canchas que contengan el deporte ingresado, y devuelve una estructura del tipo Cancha | TRAER\_CANCHAS\_POR\_DEPORTE |
| traerLocalidadPtrorFilial | idFilial | Localidad | Busca el campo localidad de la filial ingresada por idFilial y la devuelve. Este método implementa la excepción CampoVacio | TRAER\_LOCALIDAD\_POR\_IDFILIAL |
| traerTelefonoPorUsuario | Nombre, Apellido | Telefono | Busca un usuario por el nombre y apellido ingresado, y devuelve el teléfono de dicho usuario. Este método implementa la excepción NoEncontrado | TRAER\_TELEFONO\_POR\_USUARIO |
| traerTurnoPorId | idTurno | Esctructura tipo Turno | Busca un turno por el idTurno ingresado, y devuelve la estructura completa del turno | TRAER\_TURNO\_POR\_IDTURNO |
| traerTurnosPorFilial | idFilial | Lista de estructuras tipo Filial | Busca la filial por el idFilial ingresado, y devuelve todos los turnos disponibles en dicha filial | TRAER\_TURNOS\_POR\_FILIAL |

**Estructuras & Excepciones**

Se crearon las siguientes estructuras y excepcinones en Thrift

**struct Cancha** {

1: string localidad;

2: i32 idFilial;

3: string tipoCancha;

4: i32 idCancha;

5: string deporte;

}

**struct Turno** {

1: i32 idTurno;

2: string fechaHora;

3: bool vigente;

4: string localidad;

5: string deporte;

6: string tipoCancha;

7: string apellido;

}

**exception CampoVacio** {

1: string dato

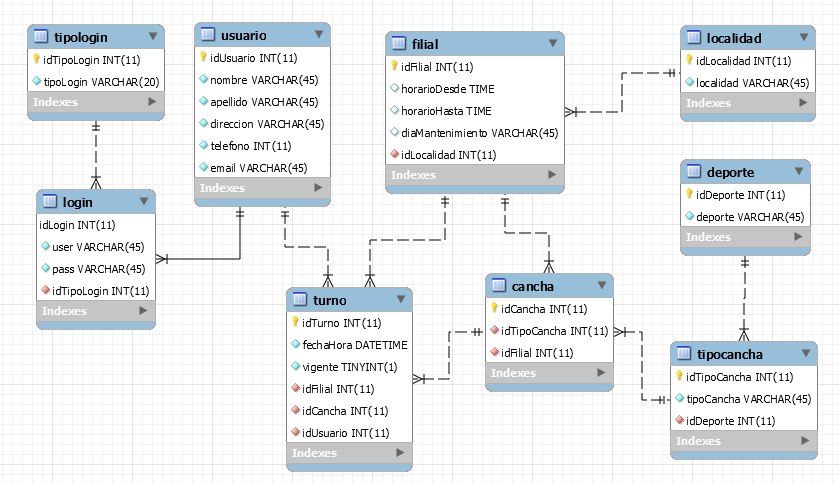
}

**exception CampoVacio** {

1: string dato

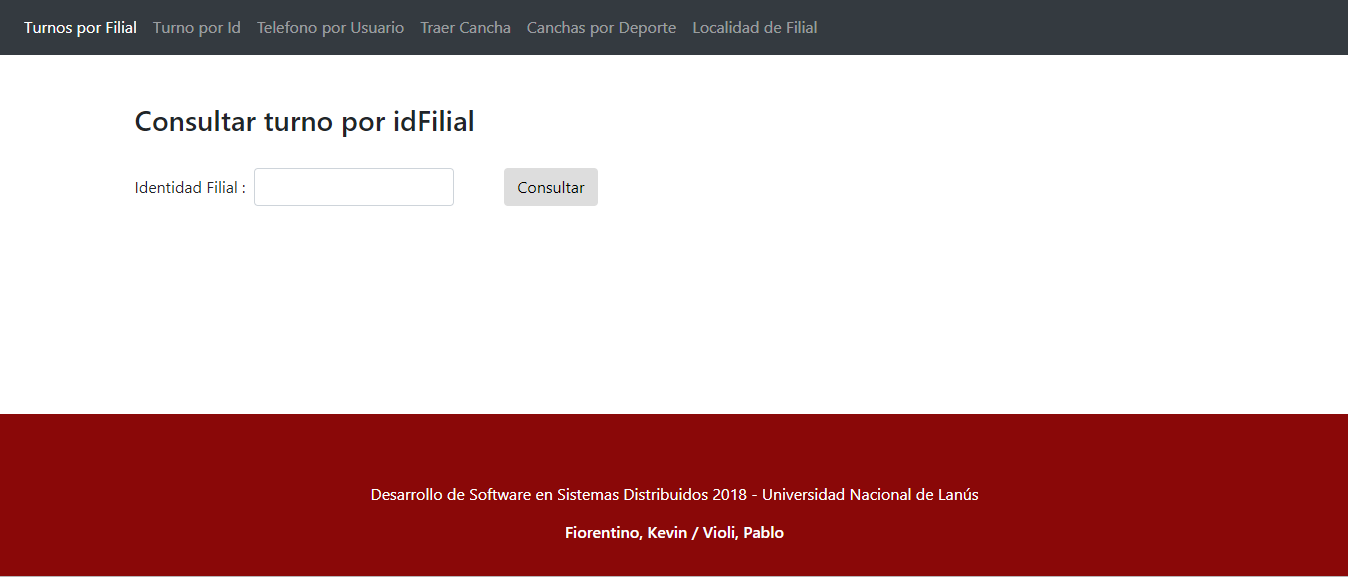
}

***Diagrama de Entidad-Relación***



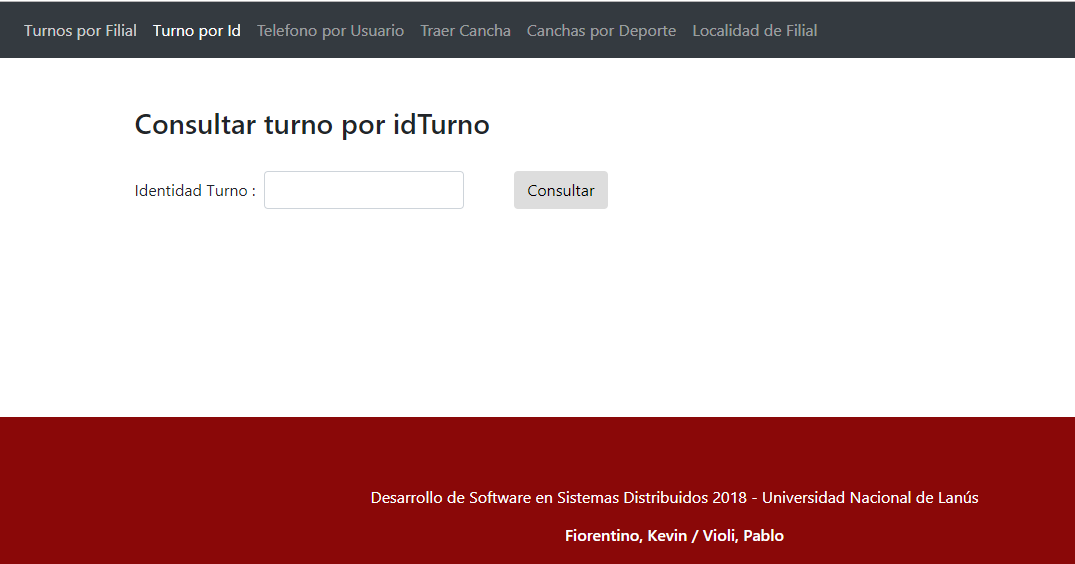
***Navegación por la aplicación y pruebas realizadas con capturas de pantalla.***

* Vista Turnos por Filial:



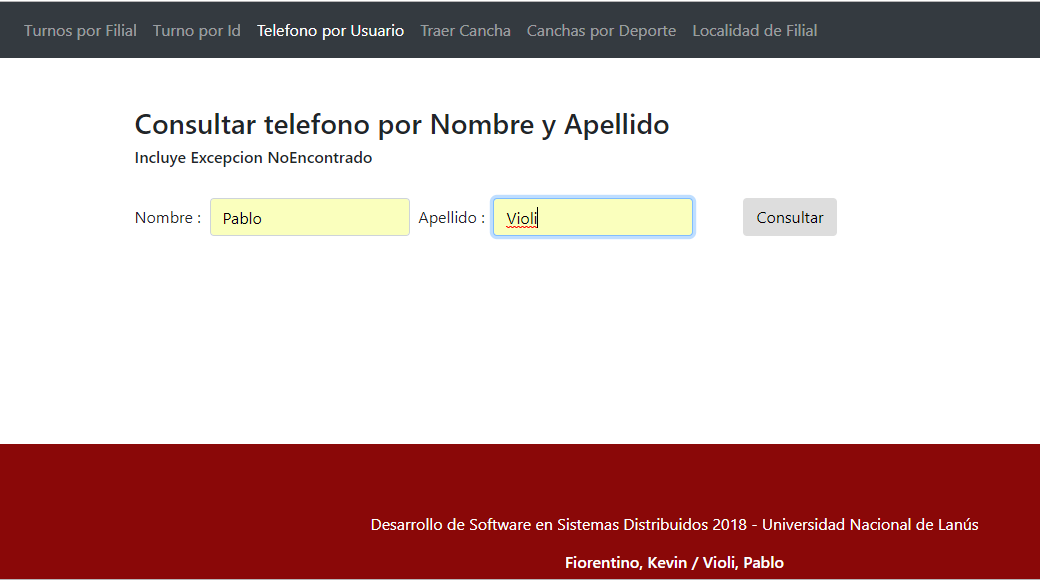


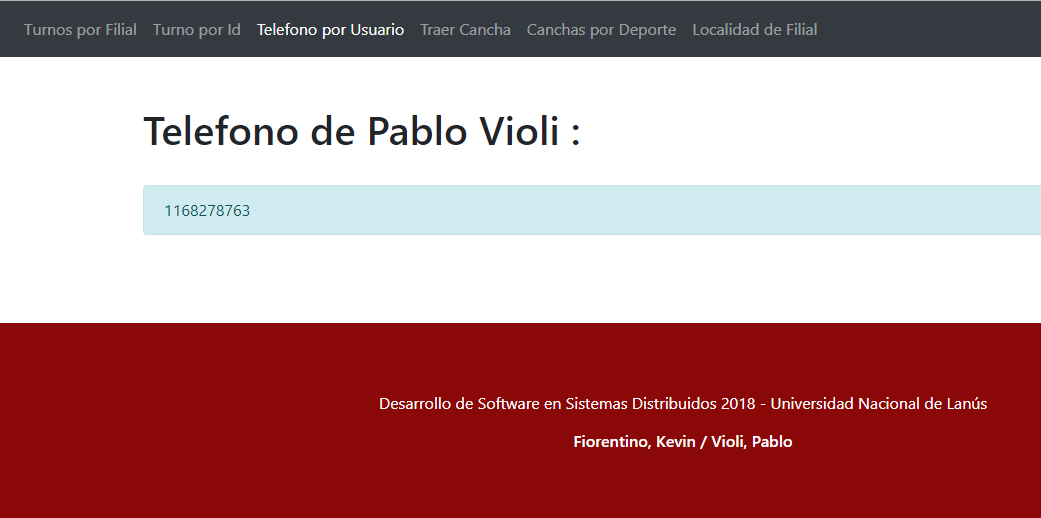
* Vista Turno por ID

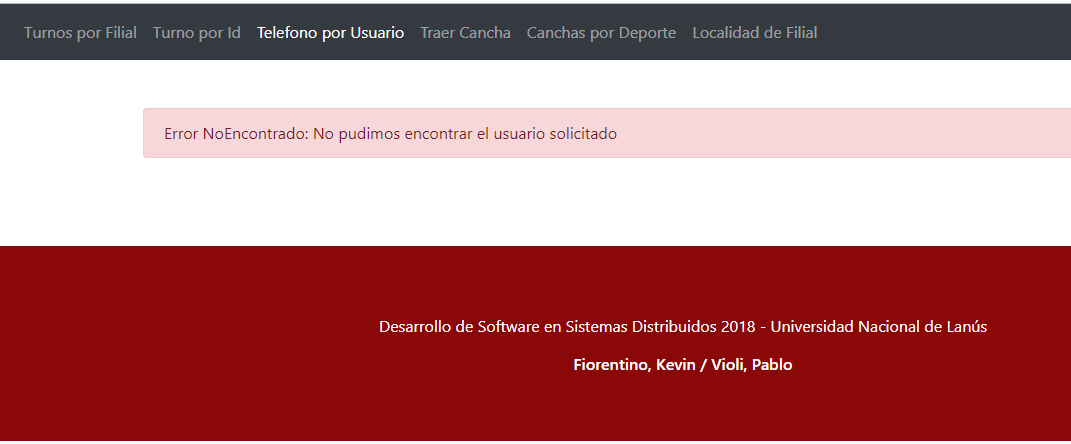




* Vista Telefono por Usuario:





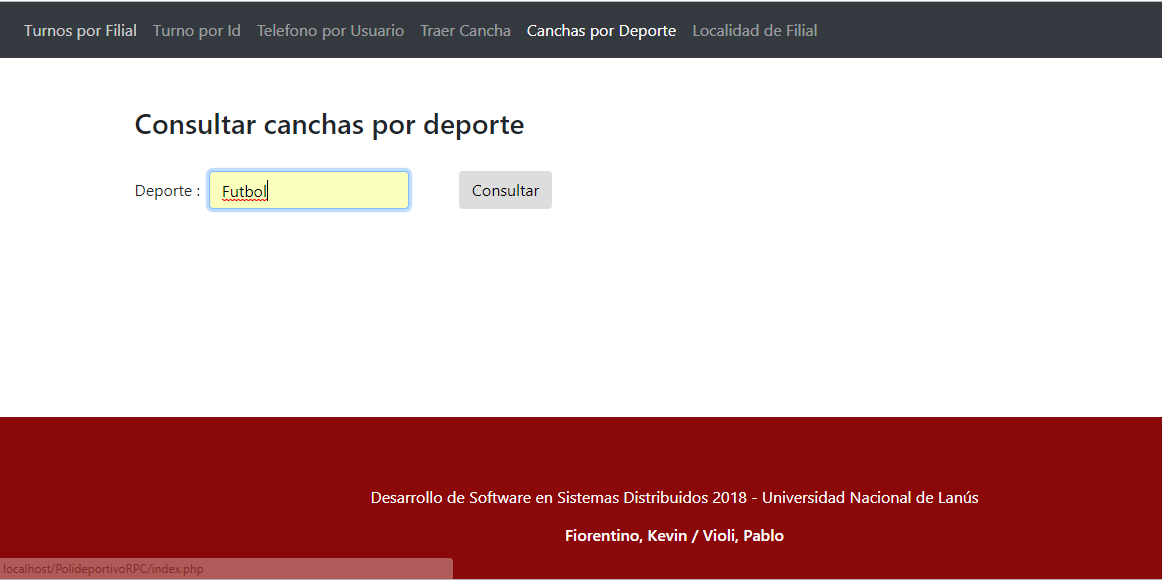
**Error NoEncontrado:** 

* Traer Cancha:





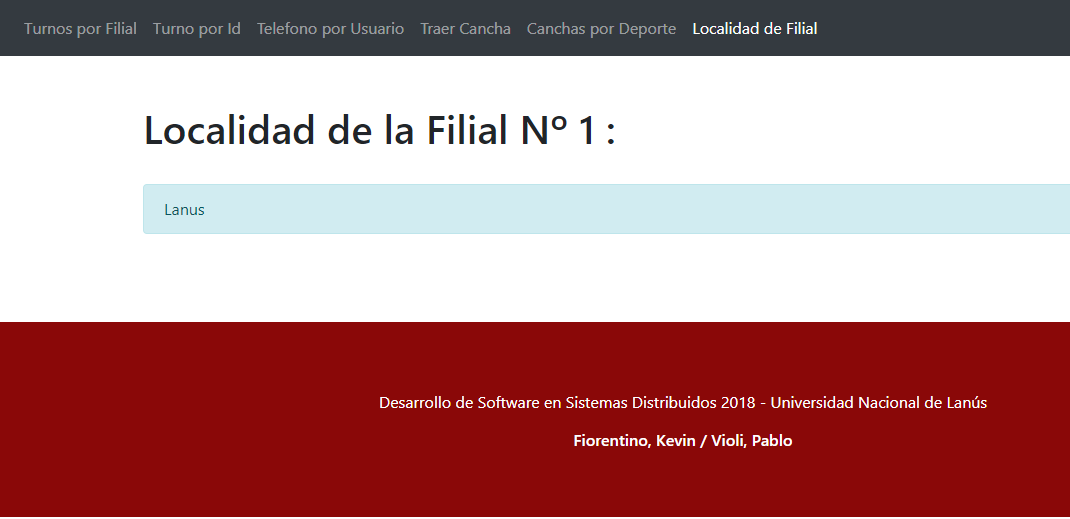
* Canchas por Deporte:





* Localidad de Filial:





**Error CampoVacio:**

