

## Evaluación II

### I. Presentación

Se le solicita la creación de una aplicación web que permita iniciar sesión, definir las vistas según usuario, realizar giros, realizar mantenciones y ver el listado de estas mismas.

#### Unidad de Aprendizaje III:

Páginas Web dinámicas con  
PHP

#### Aprendizaje esperado

Desarrolla aplicaciones Web dinámicas PHP que interactúan con bases de datos MySQL.

#### Actividades

1. Desarrollo de formulario que valide datos mediante y registre acciones en base de datos.

- **Criterios de evaluación**

Esta actividad, considera los siguientes criterios de evaluación:

- 3.1.1.- Configurando entornos para desarrollo o producción con Apache, PHP y MySQL.
- 3.1.2.- Utilizando funciones básicas del lenguaje.
- 3.1.3.- Utilizando variables de servidor, sesiones y cookies.
- 3.1.4.- Considerando diferentes métodos de envío de información.
- 3.1.5.- Aplicando operaciones de obtención, entrega, edición y borrado de datos con base de datos MySQL.
- 3.1.6.- Identificando las consecuencias y efectos de su razonamiento.

- **Puntaje total: 40 puntos.**

## II. Instrucciones

1. Utilice documentación vista en clases si así lo desea.
2. Puede utilizar frameworks para el diseño si así lo desea.
3. **Puede trabajar en grupo de hasta 3 personas.**
4. Descargue los documentos ubicados en ambiente de aprendizaje.
5. Desarrollar aplicación web según requerimientos.
6. Una vez finalizado el proyecto comprimir todo en un archivo con el siguiente nombre "evaluacionIII\_NombreAlumno.rar".
7. Subir el archivo en ambiente de aprendizaje -> Desarrollo web -> Evaluación III.

## III. Actividades

### 1. Desarrollo sitio formulario web

- a) Cree una base de datos de nombre evFinal e importe el archivo SQL ubicado dentro de los archivos descargados. **(1 puntos)**
- b) Cree el modelo correspondiente a la base de datos. Ejemplo, si existe la tabla cliente debe estar reflejada como una clase dentro de la carpeta modelo. **(5 puntos)**
- c) Cree el archivo de conexión a base de datos. **(1 punto)**
- d) Utilice jQuery para la captura de datos. **(5 puntos)**
- e) Utilice Ajax para el envío de datos. **(5 puntos)**
- f) Realice las acciones necesarias para realizar un giro. Las condiciones para ejecutar este giro son las siguientes:
  - El monto solicitado no debe superar el monto que tiene el cajero. **(2 puntos)**
  - El monto solicitado no debe superar el monto que tiene el cliente. **(2 puntos)**
  - El monto no puede ser menor a 0. **(2 puntos)**
  - Envíe los mensajes correspondientes para reflejar errores. **(2 puntos)**
- g) Una vez realizado el giro los cambios que se registran en la base de datos son los siguientes:
  - Debe insertar en la tabla giros la acción realizada con el ID del cajero al que se realizó giro, el rut del cliente que realizó giro, la fecha en la que se realizó el giro y el monto del giro. No es necesario ingresar ID en esta tabla debido a que es auto incremental. **(4 puntos)**
  - Debe realizar un update en las tablas cajero y cliente restando el monto correspondiente. **(4 puntos)**
  - Envíe los mensajes correspondientes. **(2 puntos)**
- h) Realice las acciones necesarias para realizar una recarga de dinero al cajero. Las condiciones para ejecutar esta mantención son las siguientes:
  - El monto que se carga al cajero + el monto que tenga el cajero no puede superar los 100 millones. Es decir, si el cajero tiene un monto de 90 millones el técnico solo podrá cargar 10 millones. **(2 puntos)**
  - Envíe los mensajes correspondientes para reflejar errores. **(2 puntos)**
- i) Una vez realizada la carga de dinero los cambios que se registran en la base de datos son los siguientes:
  - Debe realizar un update del monto del cajero sumado al monto ingresado. **(5 puntos)**
  - Debe insertar en la tabla mantenciones la acción realizada con el ID del cajero al que se realizó mantención, el rut del técnico que realizó la acción, la fecha en la que se realizó el giro y el monto cargado. No es necesario ingresar ID debido a que es auto incremental. **(5 puntos)**

- j) En las vistas de reporte realice lo siguiente:
- Reporte giros: Liste todos los giros realizados, además, incluya los botones ver y eliminar. El botón ver debe cargar una vista con la información completa del giro, es decir, nombre del cliente, rut del cliente, fecha, monto y el ID del cajero. El botón eliminar elimina el giro seleccionado y además realiza un update en la tabla cliente aumentando su monto en la cantidad del giro eliminada. **(5 puntos)**
  - Reporte mantención: Liste todas las mantenciones realizadas, además, incluya los botones ver y eliminar. El botón ver debe cargar una vista o modal con la información completa de la mantención, es decir, nombre del técnico, rut del técnico, fecha, monto cargado y el ID del cajero. El botón eliminar elimina la mantención seleccionada y además realiza un update en la tabla cajero disminuyendo su monto en la cantidad del giro eliminada. **(5 puntos)**
- k) Realice las acciones necesarias que permitan iniciar sesión en la aplicación según el tipo de usuario. Utilice los radios del login para diferenciar entre cliente y técnico, para ello apóyese indicando el tipo de usuario almacenado en la sesión. Debe permitir lo siguiente:
- Realizar giros: Cliente. **(2 puntos)**
  - Realizar mantenciones: técnico con tipo 1. **(2 puntos)**
  - Ver reportes y eliminar: técnico con tipo 0. **(2 puntos)**

**NOTAS:**

Para obtener la fecha automáticamente en php utilice la función date():

Para obtener todos los datos entre las tablas giros/mantención y cliente/técnico utilice la sentencia SQL JOIN.

```
SELECT * FROM giros g JOIN cliente c WHERE g.clienteRut = c.clienteRut;
```

Esta sentencia devuelve todas las columnas de giro y cliente.