

MINUTA

Minuta No. 007 de Gym Caoba

Fecha : Lunes 20 de Febrero

Hora : 9:00

Lugar : Laboratorio

Siendo las 08:00 horas del día Lunes 27 de Febrero de 2017, se reúne el grupo: **Gym Caoba**, en sesión extraordinaria previa citación realizada.

ASISTENTES:

La persona encargada que precede la reunión, Cristian Brenes G, pasó lista y determinó que había quórum.

Los asistentes a esta reunión son:

Karen Calderón C.	presente
-------------------	----------

Luis Castillo	presente
---------------	----------

Edwin Navarro B.	presente
------------------	----------

Yunen Ramos R.	presente
----------------	----------

AGENDA:

ASUNTOS:

1. Asuntos tratados y acuerdos tomados

Qué he hecho

Base de datos:

Se actualizó la tabla **tbperson**, se le agregó la llave foránea en el atributo **addressperson** con la tabla **tbaddress**.

Se creó la clase **tbhistorycampus**, la cual va a contener el histórico de personas que han ingresado en una sala a recibir cualquier servicio, contiene los siguientes atributos:

- **idhistorycampus**: Corresponde a un identificador para cada fila de información que se ingresa.
- **dnipersonhistorycampus**: Es el número de cédula del cliente.
- **idcampushistorycampus**: Es el identificador de la sala.
- **idservicehistorycampus**: Es el identificador del servicio.
- **datehistorycampus**: Corresponde a la fecha en que es ingresado el registro en la base de datos.
- **hourhistorycampus**: Corresponde a la hora en que se ingresó.

Se creó la tabla **tbpaymentmoduleclient**, la contine información para realizar la facturación. contiene los siguientes atributos:

- **idpaymentmoduleclient**: Corresponde al identificador de la tabla.
- **idpersonpaymentsclient**: Corresponde al identificador de la tabla de persona.
- **registrationdatepaymentsclient**: Corresponde a la fecha de registro de un cliente.
- **idpaymentmodulepaymentsclient**: se refiere al identificador del tipo de pago.

Se creó la tabla **tbpaymentclient** que se utiliza para almacenar los servicios pendientes de pago. contiene los siguientes atributos:

- **idpaymentclient**: Identificador de la tabla.
- **idpaymentmodulepaymentclient**: Identificador de la tabla **tbpaymentmoduleclient**.
- **idclientschedulepaymentclient**: identificador de la tabla **clientscheduleclient**.
- **paymentpaymentclient**: La cantidad de dinero a pagar de acuerdo al tipo de pago seleccionado por el cliente.
- **totalpaymentpaymentclient**: Cantidad total a pagar del servicio.

Capa domain:

se creó la clase **PaymentModuleClient** con los siguientes atributos:

- **idpaymentsclient**
- **idpersonpaymentsclient**
- **registrationdatepaymentsclient**

- **idpaymentmodulepaymentsclient**

además se agregaron los métodos accesorios

- **getIdpaymentsclient()**
- **getIdpersonpaymentsclient()**
- **getRegistrationdatepaymentsclient()**
- **getIdpaymentmodulepaymentsclient()**
- **setIdpaymentsclient(\$idpaymentsclient)**
- **setIdpersonpaymentsclient(\$idpersonpaymentsclient)**
- **setRegistrationdatepaymentsclient(\$registrationdatepaymentsclient)**
- **setIdpaymentmodulepaymentsclient(\$idpaymentmodulepaymentsclient)**

Capa data:

Se actualizó la clase **ScheduleServiceData.php**, se crearon los siguientes métodos:

- **validateCampusSchedule(\$idService,\$idDay,\$idHour):** Recibe por parámetros el identificador del servicio, el identificador del día y el de la hora. Verifica que en la base de datos no haya ya registrado un valor con la información recibida por parámetro en otras salas. Retorna un 1 (uno) si ya existe un registro y un 0 (cero) en caso contrario.
- **getScheduleByCampusHour(\$idCampus,\$idHour):** Recibe por parámetros el identificador de la sala y el de la hora. Retorna el identificador del servicio que se imparte en los valores recibidos por parámetros, si no existiera un servicio en ese momento entonces se retorna un 0 (cero).
- **existSessionInCurrentCampus(\$dniClient,\$date,\$idHour):** Recibe por parámetros el número de cédula del cliente, fecha y el identificador de la hora. Retorna un 1 (uno) si ya el cliente se encuentra registrado en la base con la fecha y hora recibida por parámetro, y un 0 (cero) en caso contrario.

Se creó la clase **HistoryCampusData.php**, contiene la siguiente información:

- **Atributo:**
 - **connection:** Atributo que hace referencia a la clase que permite crear la conexión con la base de datos.
- **Métodos:**
 - **constructor:** Método que inicializa los valores globales.
 - **existPerson(\$dniPerson):** Recibe por parámetro el número de cédula de un cliente. Retorna un 0 (cero) si el cliente no existe en

la base de datos, y un 1 (uno) en caso contrario.

- **registerClientInCampus(\$dniClient,\$idCampus,\$idService,\$date,\$idHour):** Recibe por parámetros el número de cédula de un cliente, el identificador del campus, del servicio, la fecha actual del sistema y el identificador de la hora. Ingresa los valores recibidos por parámetros en la tabla **tbhistorycampus**.
- **getLastID(\$table):** Recibe por parámetro el nombre de la tabla que se desea consultar, retorna el próximo identificador que se puede asignar en la tabla consultada.

creó la clase **PaymentModuleClientData** con los siguientes métodos:

- **insertPaymentModuleClient(\$PaymentModuleClient):** Se utiliza para almacenar la información en la tabla **tbpaymentmoduleclient**.
- **insertPayment(\$idPymentModule, \$idclientschedule, \$payment, \$totalpayment):** Método utilizado para almacenar en la base de datos en la tabla **tbpaymentclient**.
- **getPaymentModule(\$idPerson):** Método utilizado para obtener el tipo de pago elegido por el cliente cuando se realizó el registro.
- **getMaxId():** Se utiliza para obtener el máximo identificador de la tabla **tbpaymentmoduleclient**.

Capa business:

Se creó la clase **HistoryCampusBusiness.php**, el cual permite la interacción entre la vista que contiene los dispositivos detectores de código QR y la clase **HistoryCampusData.php**. Incluye las clases: **HistoryCampusData.php** y **ScheduleServiceData.php**.

Capa presentation:

Se actualizaron las clases: **createservice.php**, **cancelservice.php**, **updateservice.php** y **renewservice.php**, con el fin de hacer uso de la función **datepicker()** del paquete **jquery-ui.1.12.1**.

Se creó la clase **logincampus.php**, el cual simula el dispositivo detector de código QR que se implementaría en la entrada de cada una de las salas. Hace uso de métodos de los paquetes: **ion.sound-3.0.7** y **html5-qrcode**.

JS:

Se creó la clase **GetScheduleCampus** con el fin de retornar todos los servicios que ofrece el gimnasio en una determinada sala.

También se implementó la clase **GetScheduleService** que nos retorna el horario de un determinado servicio.

Capa presentation:

Se reestructuró la clase ShowSchedule para mostrar y al mismo tiempo el usuario tenga la facilidad de escoger uno o varios servicios al mismo tiempo.

En la clase person e inventory se le agregó el datepicker a las fechas.

Se reestructuró la clase person para mostrar el código qr cuando se registra un cliente, además se planteó de forma dinámica el poder realizar la actualización, eliminación y registro de un barrio.

Además se creó la clase invoice para que el usuario pueda observar la factura y realizar el pago de los servicios contratados.

Resources:

Se agregaron los siguientes paquetes:

- **html5-qrcode:** Permite la lectura de códigos QR, mediante una cámara.
- **ion.sound-3.0.7:** Permite reproducir sonidos dinámicamente.
- **jquery-ui.1.12.1:** Conjunto de elementos que mejoran el aspecto visual de la página, se empleó para los datepicker.

a

Qué voy a hacer.

Impedimentos

Tareas realizadas que no fueron planeadas.

2. Asuntos Pendientes

Llevar el registro del ingreso por servicio contratado.

3. Asuntos Nuevos

Módulos funcionales nuevos que a la luz de las discusiones van siendo pertinentes de desarrollar, aún cuando no necesariamente lleguen a implementarse. No comprometen un hito de desarrollo

4. Cumplimiento de tareas asignadas

Tarea	Responsable	Grado de conclusión	Nota
Horario semanal con servicios divididos por sala y validar que no choquen	Luis Castillo	Completado	
Manejo de ingresos diarios	Luis Castillo	Pendiente	
Reporte de morosidad	Luis Castillo	Pendiente	
Realizar la facturación	Karen Calderón	Pendiente	
Enviar al cliente el código QR	Karen Calderón	Completado	
Máscara de fechas y datepicker	Karen Calderón	Completado	
Sustituir el username por el correo del cliente, y hacerlo atributo único	Karen Calderón	Completado	
Agregar barrio sin recargar	Karen Calderón	Completado	
Recuperar contraseña desde el correo.	Karen Calderón	Completado	
Actualizar el árbol genealógico.	Karen Calderón	Completado	
Encargada de minuta	Karen Calderón		
Lector de código QR para cada sala, con su ingreso en la tabla tbhistorycampus .	Yunen Ramos Ramírez	Completado	
Implementar datepicker en las clases: createservice , cancelservice y renewservice .	Yunen Ramos Ramírez	Completado	
Validar que un servicio no se puede dar dos veces a la misma hora en distintas salas.	Yunen Ramos Ramírez	Completado	
Pasar los valores quemados de la base a español.	Yunen Ramos Ramírez	Completado	
Pasar a español las vistas de: servicio, horarios salas y el correo de cancelación del servicio.	Yunen Ramos Ramírez	Completado	
Agregar la llave foránea a la tabla tbperson , específicamente el atributo addressperson con la tabla	Yunen Ramos Ramírez	Completado	

tbaddress.			
Recomendación según rendimiento	Edwin Navarro	Completado	
Cálculo de niveles de rendimiento	Edwin Navarro	Completado	
Criterios para métricas	Edwin Navarro	Completado	
Periodicidad de medidas	Edwin Navarro	Completado	
Porcentaje de avance por cada métrica	Edwin Navarro	Completado	
Cambios de criterios	Edwin Navarro	Completado	
Validaciones en tomar medidas	Edwin Navarro	Completado	
Gráfico de métricas en un histórico en el tiempo para el cliente	Edwin Navarro	Completado	

*Aunque el proyecto es grupal, en cada iteración se asignará una persona del grupo como responsable de cada una de las tareas asignadas.

**Existen tareas ya definidas como avances revisables en cada iteración:

1. Diagramas de casos de uso
2. Modelo de datos
3. Diagrama de aplicación
4. Módulos funcionales en ejecución

PRÓXIMA REUNIÓN:

Fecha: ---

Hora: 8:00 AM

Lugar: Laboratorio C

CLAUSURA:

Cristian Brenes G, como persona encargada de preceder la reunión, clausura la reunión a las XXXX horas.