24 de junio de 2014

Sistema de Administración Gimnasio BJ

Proyecto Diseño de Software

Integrantes:

Braulio Alpízar Morales
Leslie Becerra Blanco
Yader Morales López

ADMINISTRACIÓN DE TECNOLOGÍA DE INFORMACIÓN

Contenido

Resumen ejecutivo

En el siguiente documento se presenta una breve descripción de las funcionalidades de la aplicación, así como las tecnologías empleadas e interacción con sistemas externos.

Se trata de una aplicación web para un gimnasio, la cual va a tener dos enfoques:

- El primero será hacia la parte administrativa del gimnasio, en donde se contará con dos tipos de roles para el acceso y visualización de la información.
    - El primer rol será el de administrador, el cual será capaz de insertar, editar, borrar y consultar los clientes, esta última por una serie parámetros como: Nombre, Apellido, Carnet, Rango de Fechas, Morosidad, etc. También será capaz de editar horarios, agregar usuarios y asignarles un rol y otras funcionalidades ligadas a los clientes y de la información propia del gimnasio.
    - El segundo rol será el de Instructor, el cual se encargará de insertar, editar, eliminar rutinas y medidas por cada cliente, también será capaz de crear, editar y eliminar ejercicios, los cuales son agregados a las rutinas dependiendo del objetivo de las mismas, también podrá realizar consultas sobre los clientes por los mismos parámetros que el rol administrador.
- La segunda funcionalidad de la aplicación web será enfocada hacia los clientes, en donde estos podrán visualizar sus rutinas, información de pago, avisos, eventos, horarios, información general, sugerencias de dietas alimenticias para complementar el ejercicio y editar información de su perfil.

Esta aplicación se desarrollará con el Framework de .Net 4.5.1, también se hará uso Azure (PaaS) para la publicación de la página Web y para manejar la base de datos en la nube, la cual utiliza el motor SQL-Server, el sistema también contará con un Web Service con el cual la aplicación móvil multiplataforma diseñada para el gimnasio pueda conectarse e interactuar con la base de datos y demás funcionalidades.

Para obtener cierto tipo de información y también para facilitar el acceso al cliente, la aplicación va a interactuar con dos sistemas externos:

- Yummly el cual va a proveer información sobre la alimentación apropiada para complementar el ejercicio.
- Facebook en donde podrá ingresar a través de su cuenta.

Especificación de requerimientos

Funcionalidades Administrativas

- El sistema deberá tener tres tipos de roles: Usuario administrador, usuario instructor y los clientes.
- El rol de administrador puede realizar CRUD de usuarios, la creación de usuarios tendrá los siguientes datos, nombre, apellidos, cédula, edad, sexo, correo electrónico y teléfonos.
- El usuario que tenga un rol administrador, puede asignar los roles de administrador e instructor.
- El sistema deberá permitir que rol de administrador y el de instructor pueden asignar a un usuario el rol de cliente el cual además de los datos normales de usuario, mantendrá una fecha de ingreso y las medidas corporales iniciales (Triceps, Biceps, Espalda, Cintura, Cadera, Cuadriceps, Pantorrillas, Pecho, Peso, Altura, Índice de Masa Corporal, Muñeca, Codo, Rodilla.
- El sistema deberá permitir la autenticación de los usuarios en sus distintos roles.
- El sistema debe permitir al rol de tipo administrador crear, editar y eliminar eventos del gimnasio, cada evento tiene los siguientes datos: Nombre, descripción, día, hora e imagen del evento.
- El sistema debe permitir al usuario de tipo administrador enviar avisos importantes (por ejemplo que el gimnasio no abra un día específico o que el horario sea modificado) a los clientes del gimnasio, cada aviso tiene la siguiente información: Nombre, descripción, fecha y hora.
- El sistema debe permitir al rol de tipo administrador consultar los clientes del gimnasio de acuerdo a los siguientes parámetros: Nombre, apellidos, clientes morosos, fecha de ingreso y por rangos de fechas.

Funcionalidades intermedias

- El sistema debe permitir al usuario de tipo instructor registrar cada cierto tiempo (Mes, semana) las medidas del cliente del gimnasio (medidas corporales en cm).
- El sistema deberá permitir al rol de tipo instructor crear, editar y eliminar rutinas, cada rutina tiene los siguientes datos: Nombre, objetivo y descripción.
- El sistema deberá permitir al rol de tipo instructor crear, editar y eliminar ejercicios, cada ejercicio tiene los siguientes datos: Nombre, descripción, equipo a utilizar, sets, repeticiones, peso, ajuste, tiempo, tiempo de descanso.
- El sistema deberá permitir al rol de tipo instructor añadir y eliminar ejercicios a una rutina específica.
- El sistema deberá permitir al rol de tipo administrador e instructor consultar los clientes, por nombre, carnet, apellido. Este con el fin de determinar la entrada de clientes al gimnasio.
- El sistema deberá permitir al rol de tipo administrador e instructor registrar la entrada de un cliente al gimnasio.

Funcionalidades externas

- El sistema deberá mostrar la siguiente información: Nombre del gimnasio, descripción, misión, visión, encargado, información de contacto (email y número de teléfono), horario de atención y la ubicación de las instalaciones.
- El sistema deberá permitir a los tres tipos de roles modificar su información personal, se permitirá modificar los siguientes datos: Ocupación, teléfono, email y una fotografía.
- El sistema deberá permitir al rol de tipo cliente seleccionar la rutina que desee realizar.
- El sistema deberá permitir al rol de tipo cliente seleccionar el ejercicio que desee realizar según cada rutina.
- El sistema deberá permitir al rol de tipo cliente visualizar las instrucciones de cada ejercicio.

- El sistema deberá mostrar al rol de tipo cliente, avisos agregados por el administrador del gimnasio.
- El sistema deberá mostrar al rol de tipo cliente, los eventos que tiene el gimnasio durante la semana.
- El sistema deberá recordar al rol de tipo cliente, la próxima fecha de pago.

Prioridades y requerimientos no funcionales

En este caso, la priorización se da en tres niveles, de acuerdo a lo expuesto en el apartado de especificación de requerimientos:

1) Funcionalidad Administrativa: En este caso, se van a manejar como requerimientos indispensables para la aplicación, todos los englobados en esta categoría de prioridad 1.
2) Funcionalidad Intermedia: Los requerimientos que componen esta prioridad, serán los relacionados con el rol de instructor, se catalogan como prioridad 2
3) Funcionalidad Externa: Estos requerimientos se componen de los ejecutados por el rol de cliente en la aplicación, su prioridad es 3 debido a que no son parte del núcleo, sino más una consulta a los datos y una muestra de los mismos.

Requerimientos no funcionales de la aplicación

Confianza

- El sistema contará con un método de autenticación basado en un carnet y una contraseña para mantener privado el perfil de cada usuario, donde por medio de cookies se manejará el acceso a las páginas que le corresponden al rol, además de protocolo al ingresar el usuario y contraseña de HTTPS.

Rendimiento

- El tiempo de respuesta de la página web dependerá de la conexión del cliente, pero se tolera el retraso de un 1% sobre este valor.

Mantenimiento

- La aplicación deberá tener documentación interna que describa las funcionalidades de cada clase, así como los métodos usados.
- La aplicación deberá tener documentación que describa las decisiones de diseño tomadas con respecto a manejo de interfaz, lógica y datos, además de los patrones utilizados y de las cuentas que se manejan para Windows Azure.

Usabilidad

- La autenticación del rol cliente se realizará mediante un número de carnet y una contraseña suministrada por el gimnasio.
- El sistema debe mostrar al usuario la ubicación de las instalaciones mediante un mapa.
- La aplicación deberá mostrar al usuario los eventos que tiene el gimnasio durante la semana mediante un horario semanal ordenado por días (Columnas) y horas (Filas).
- El tiempo de aprendizaje sobre el uso de la aplicación para los clientes no debe superar los diez minutos.

Descripción de diseño de alto nivel

En esta sección, se especifican los diagramas que poseen un nivel de abstracción alto, tal es el caso del diagrama de arquitectura, paquetes, componentes y despliegue, para el sistema de gestión del gimnasio BJ.

Diagrama de arquitectura conceptual

En este caso, se busca reutilizar soluciones y fomentar el uso de componentes independientes, por ello se diseña este diagrama pensado en el patrón de n-capas y cliente-servidor. El fin de esta implementación es manejar versatilidad en la petición de datos a las capas superiores e inferiores, como también la descarga de procesamiento en el componente del cliente.

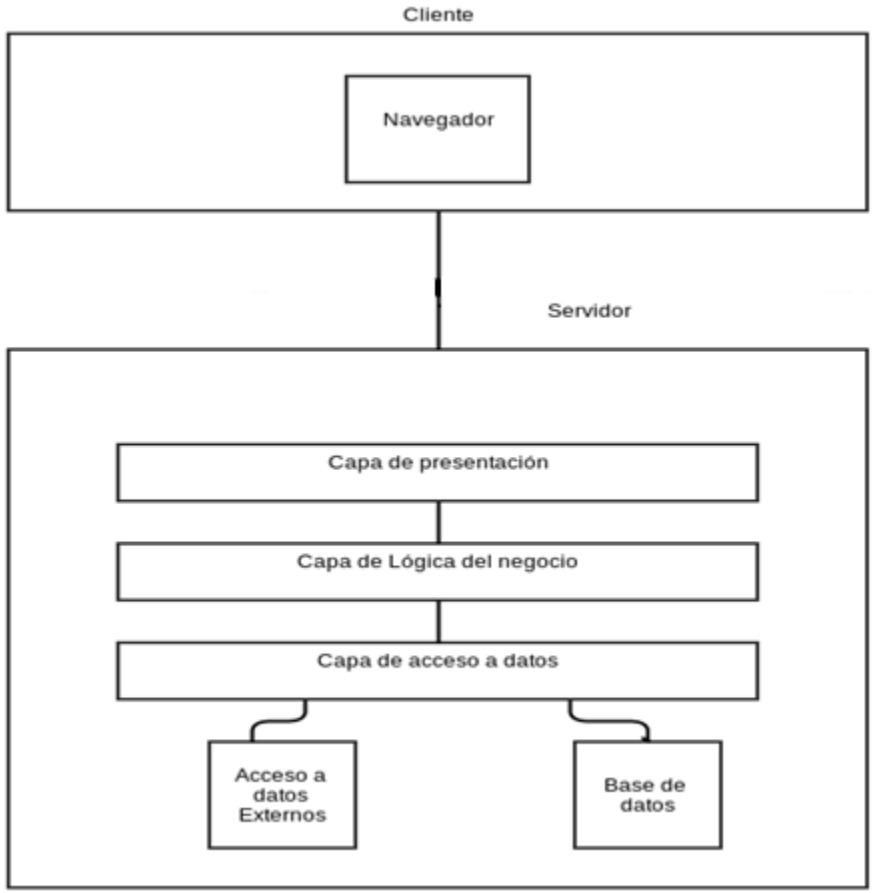


Diagrama de paquetes

Este diagrama muestra la organización de los elementos que componen el sistema a gran escala, donde se procura con un nivel de abstracción considerable mantener coherencia en el diseño. Las dependencias y los subsistemas que lo componen son también parte fundamental de esta vista.

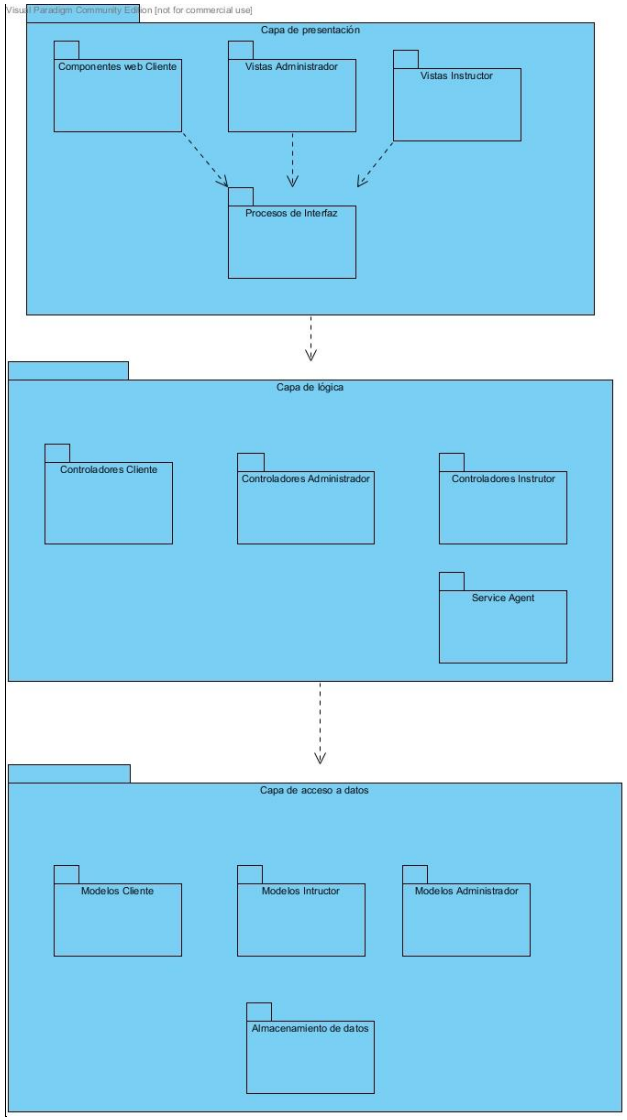


Diagrama de componentes

En este diagrama se puede visualizar la relación existente entre los diversos componentes del sistema, en este caso tenemos tres principales, el navegador, el Backend, donde se manejará la lógica y el procesamiento de datos emitidos por las distintas vistas y los API externos, en este caso enlazados con el Backend directamente por una interfaz.

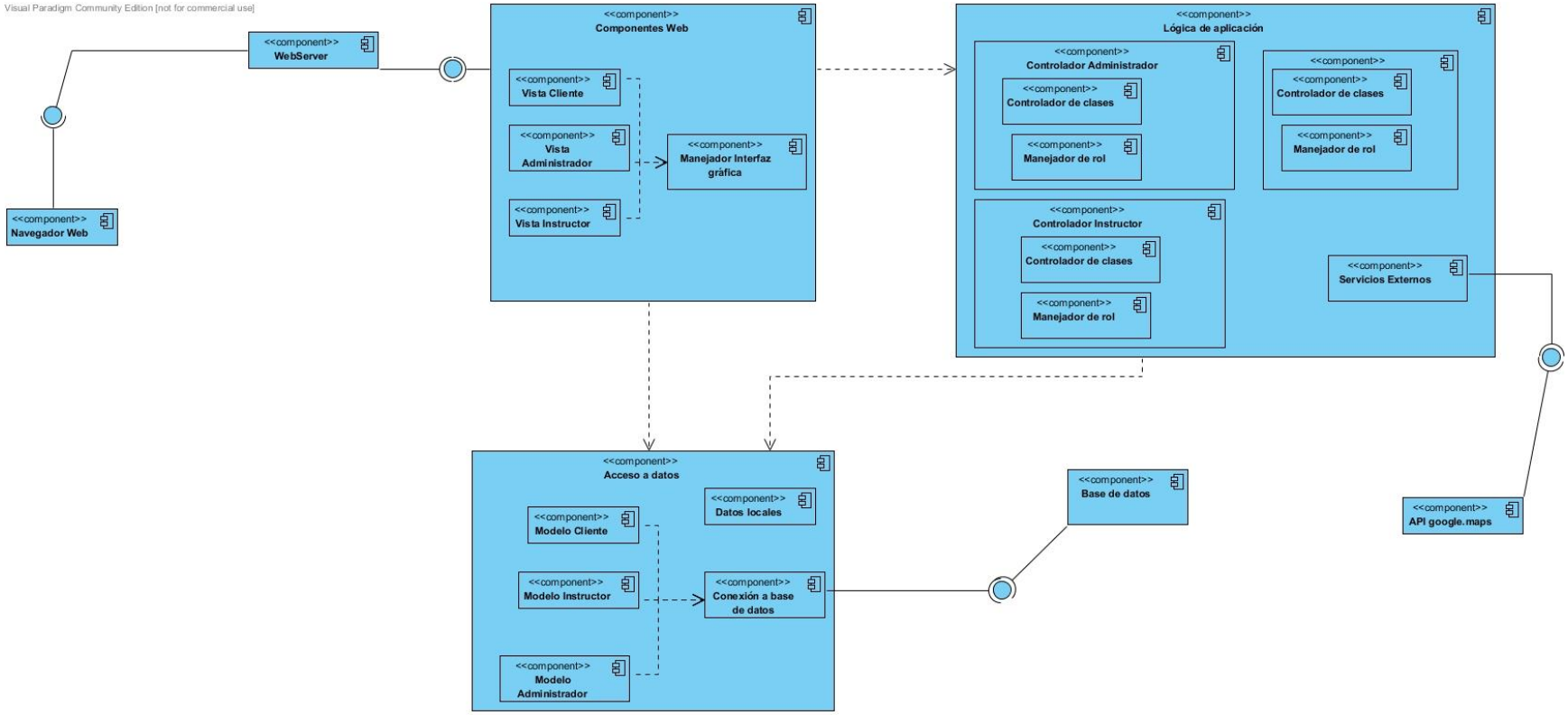
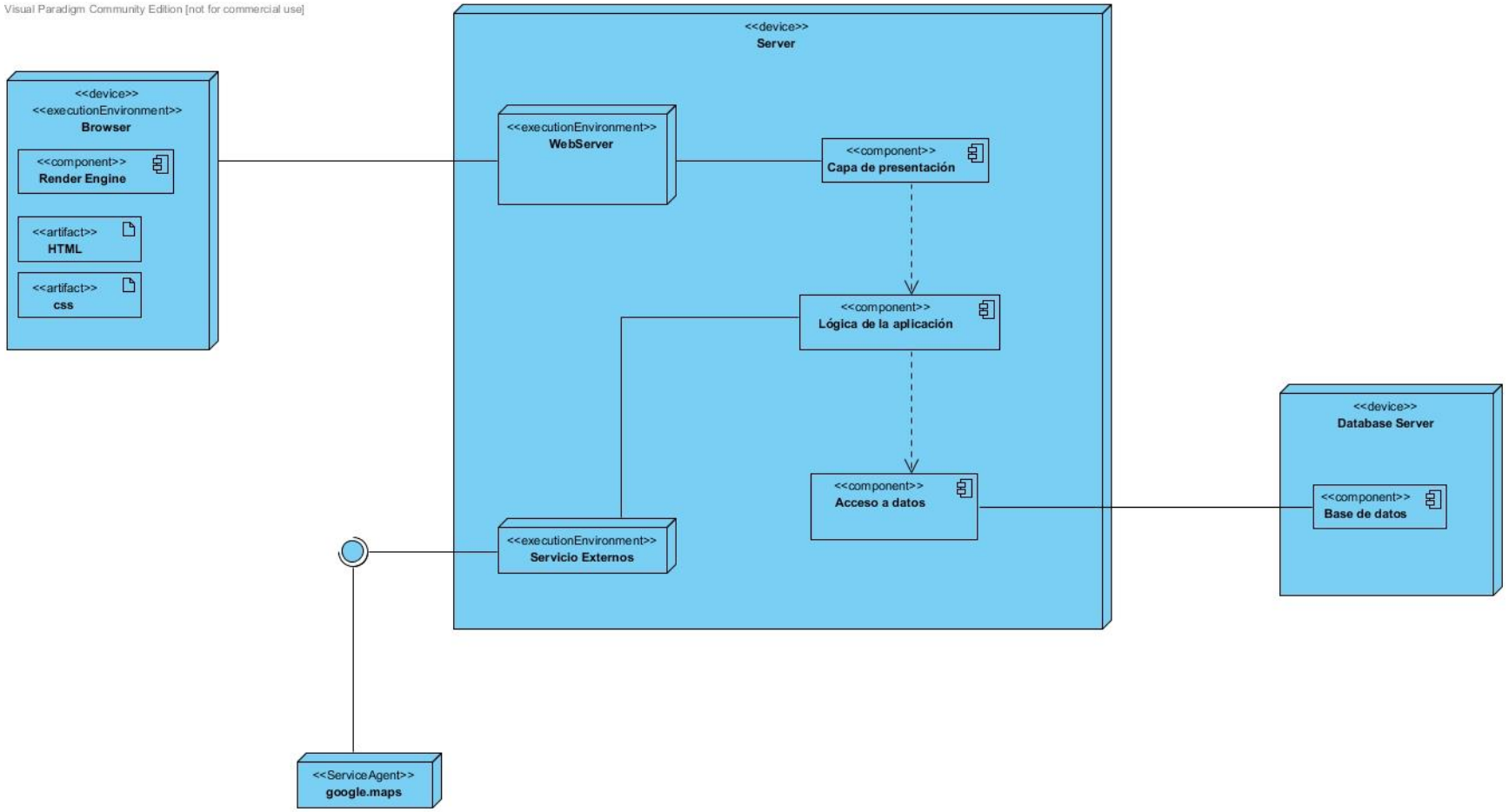


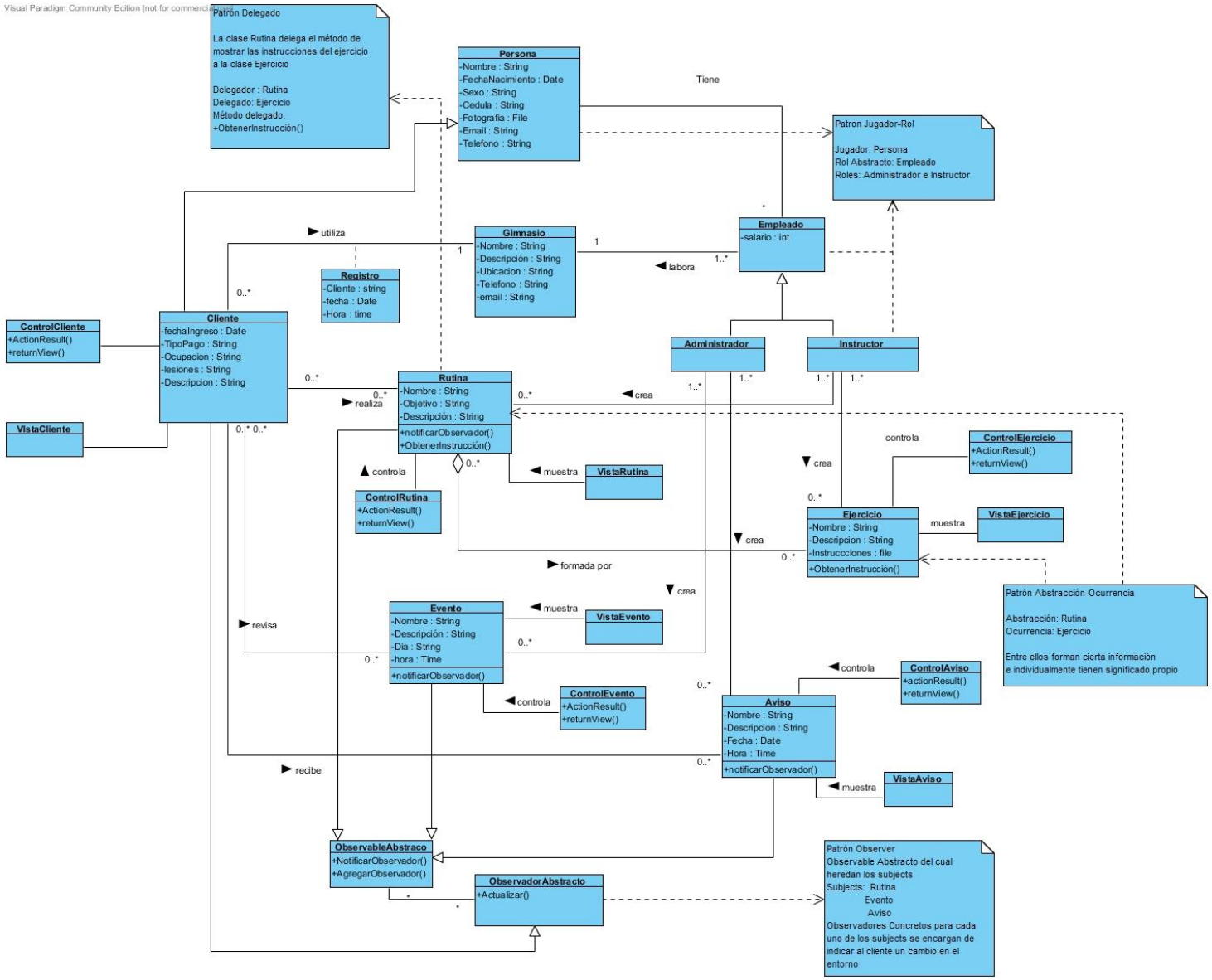
Diagrama de despliegue

En este caso, similar al diagrama de componentes, podemos visualizar de manera conceptual las distintas capas físicas que posee el sistema, donde un dispositivo, es el encargado de tomar los datos necesarios para el procesamiento del Backend, en este caso un servidor que se encuentra en la nube por medio del servicio de Windows Azure y también la comunicación que se maneja con API externo, ya sea por JSON o por REST.



Descripción detallada

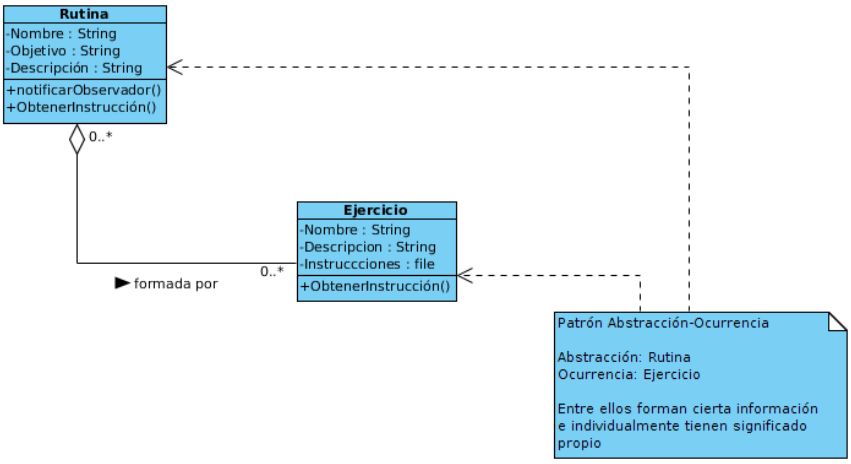
Esta sección representa el diagrama de clases, el cual posee un nivel de abstracción menor, permitiendo así la visualización de los métodos y atributos a utilizar, además de los distintos patrones.



Patrones usados

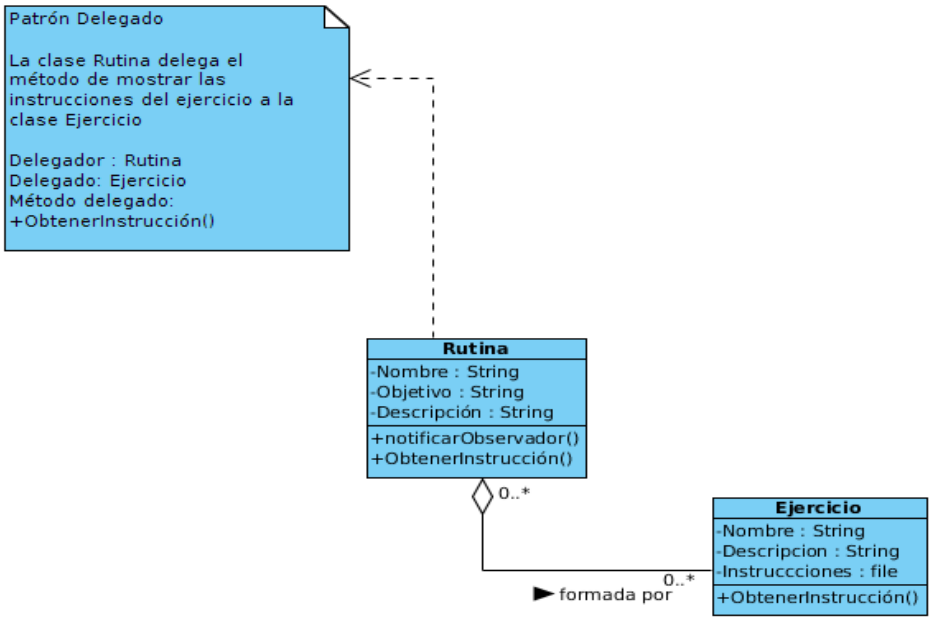
Abstracción-Ocurrencia

El patrón Abstracción ocurrencia se implementa ya se tiene una rutina y por otro lado ejercicios. La rutina tiene ciertos valores generales pero una vez asignados ciertos ejercicios (de 0 a *), se convierte en una ocurrencia de determinada rutina con ciertos ejercicios asignados



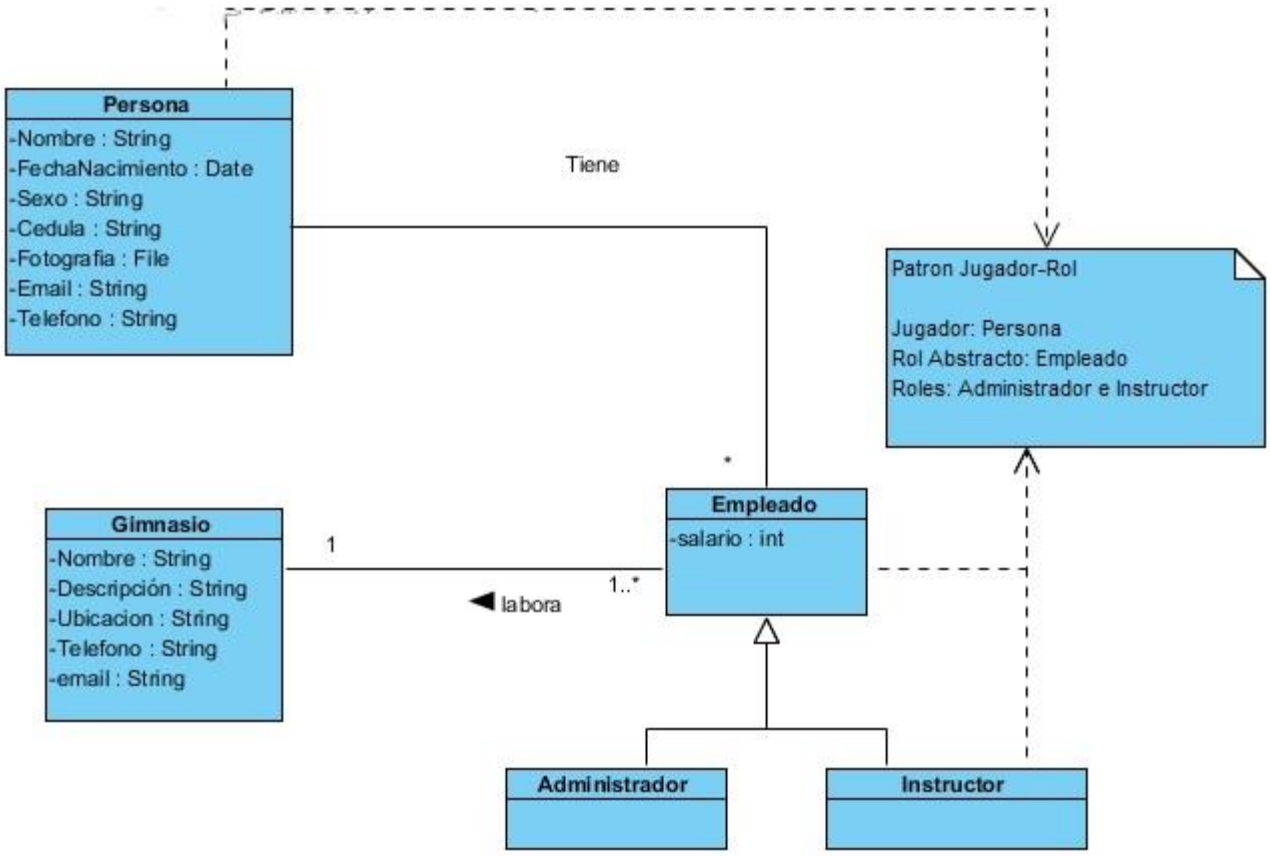
Delegado

En este patrón la clase rutina está conformada por cierta cantidad de ejercicios, los cuales a su vez tienen instrucciones, debido a que la clase rutina no puede mostrar las instrucciones de cada ejercicio ya que no pertenecen a su clase, lo delega a la clase Ejercicio mediante el método ObtenerInstruccion().



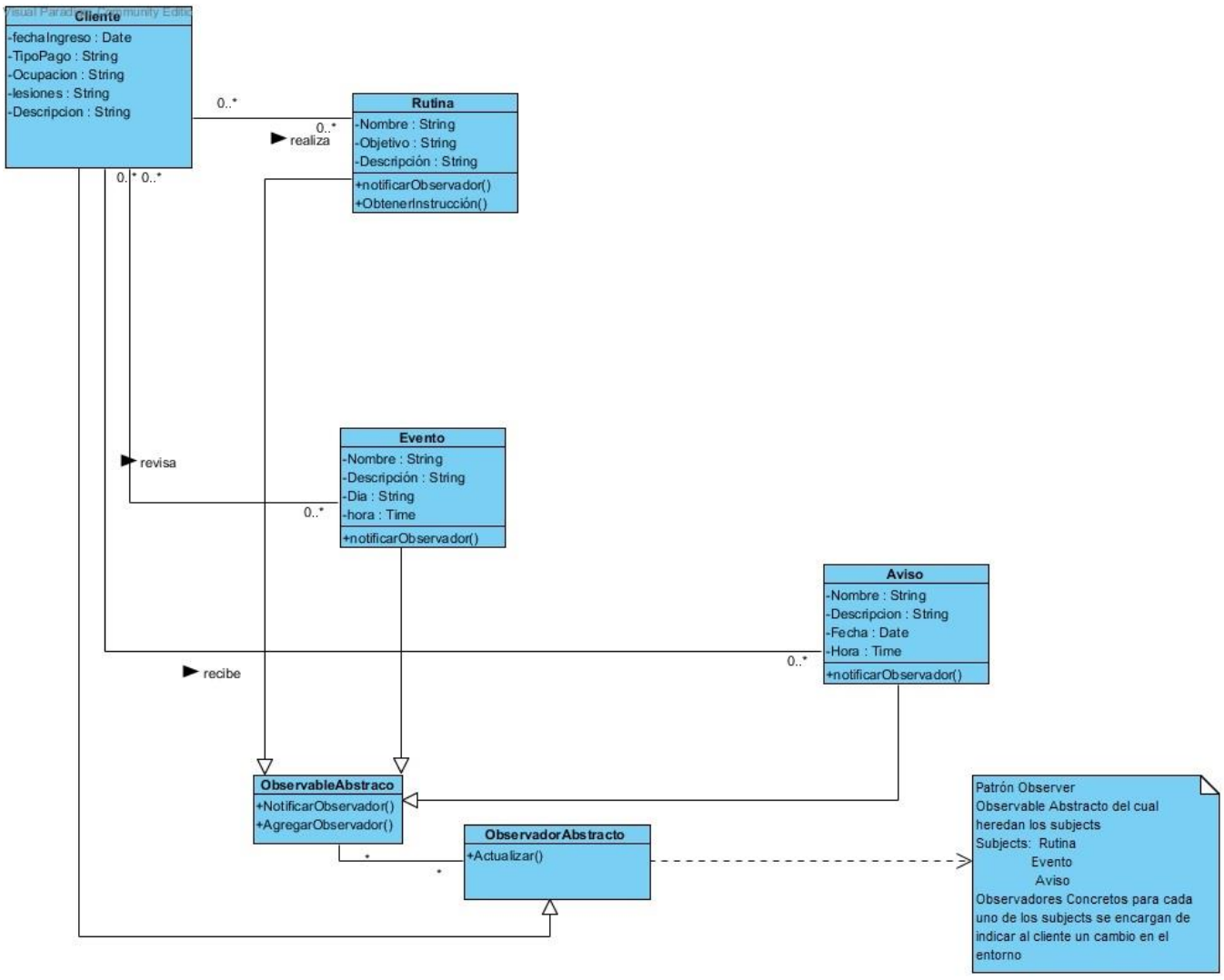
Jugador-Rol

Es utilizado por la clase Usuario la cual es abstracta y las clases administrador e instructor, dicho patrón es utilizado para el resolver el problema que se presentaba que un usuario del gimnasio puede ser administrador e instructor, por lo cual implementado un rol abstracto Usuario permite que un usuario tenga 1 o más roles en el gimnasio.



Observador

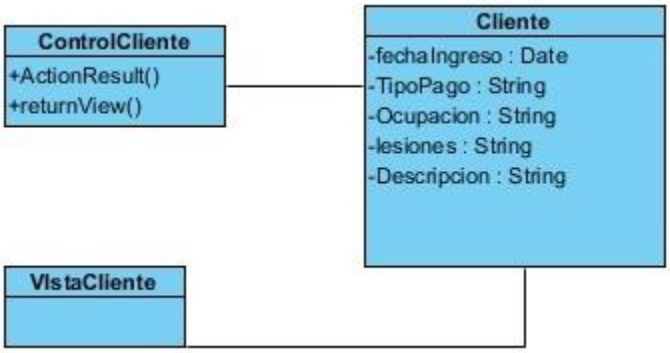
El patrón Observer es utilizado en el proyecto por las clases abstractas ObservableAbstracto, ObservadorAbstracto y por las clases concretas ObserverAviso, ObseverRutina y ObserverEvento. La clase ObservableAbstracto es implementada debido a la multiplicidad de información que debe ser mostrada, debido a que las clases Rutina, Evento y Aviso van a mostrar información, por lo que la clase ObservableAbstracto unifica esta información y a su vez esta notifica a la clase ObservadorAbstracto la cual posee un método que se va a implementar en las clase concretas ObserverAviso, ObserverRutina y ObserverEvento para ya poder desplegar la información para las distintas plataformas.



MVC

La aplicación utilizará el patrón MVC 5 proporcionado por asp.net bajo el Entity Framework, el cual lo implementa de la siguiente manera:

- **Modelo:** El Entity Framework utiliza la clase DBContext para crear objetos a partir de la base de datos que está siendo utilizada, esta se encuentra definida en la hoja de configuración de la aplicación como un acceso a datos.
- **Vista:** se implementan por medio de HTML 5 con una hoja de características asociada a cada una de las vistas.
- **Controlador:** las clases generadas a partir del modelo, se encuentran heredando de la clase Controller de MVC, por el cual se mantienen las peticiones desde las vistas y las respuestas del modelo.



Interacción con sistemas externos

La aplicación a desarrollar utilizará un sistema externo para realizar la funcionalidad de presentar donde se encuentra ubicado el gimnasio, para la cual se conectará con:

- Google Maps: este servicio ofrece un API el cual se utilizará para mostrarle al usuario en un mapa donde está ubicado el gimnasio y el usuario pueda llegar a este.

Reglas del negocio

- Cada usuario de tipo cliente puede tener asociada más de una rutina
- Un usuario puede tener muchos roles

Otros detalles

- La aplicación se encuentra en la página www.gimnasiobj.net