Documento de Arquitectura de Software

Grupo 4

Romina Lopez

Cintia Leal

Franco Cuesta

**Contenido**

[**1 Introducción.**](#_ax89rj3qs9jg) **3**

[**2 Vista de Casos de Uso**](#_clnuhjrkjmif) **3**

[2.1 Casos de uso identificados](#_m6zukwkww5nl) 3

[3 Diagrama de Casos de Uso Críticos](#_ikd9onqnhhkt) 4

[3.1 Actores](#_ffkjz7r7o748) 5

[3.1.1 Usuario](#_99uyqlt28fv4) 5

[3.1.2 Administrador](#_ep009allyed1) 5

[3.2 Especificación de Casos de Uso Críticos](#_ogyc4i4muhjv) 5

[5 Subsistemas](#_nqibd7xxeazx) 6

[5.0.1 Business Layer.](#_t7hf09dd6jzn) 6

[5.0.1.1 Diagrama de Componentes.](#_omc1x32lt5bl) 6

[5.0.1.2 Descripción.](#_qjjgywxpzj59) 7

[**6 Vista de Distribución (Deployment).**](#_95z9vb2svujc) **9**

[6.1.2 Nodos](#_nl824clcdv5e) 10

[6.1.3 Conexiones](#_qu2rlygbd43d) 10

[**7 Vista de Implementación.**](#_7lcu0nv45doc) **10**

# **1** **Introducción.**

TuPenca es una plataforma informática para realizar pencas deportivas. La idea de una penca es poder sumar puntos por acertar resultados en eventos deportivos. Dichos eventos deportivos suelen formar parte de uno o más torneos o campeonatos. El público objetivo de este producto es, por lo general, fanáticos de determinados deportes, empresas y/o grupos de personas que les interesa vivir la competencia con este aliciente adicional. Cabe mencionar que todos los tipos de deportes pueden tener su penca la cual varía en la forma de asignar ganadores y el puntaje por acertar.

El sistema cuenta con dos módulos, uno de front office, de acceso público, y otro de back office para la administración del sistema, con autenticación restringida para un grupo reducido de administradores.

# **2** **Vista de Casos de Uso**

## 2.1 Casos de uso identificados

Módulo Front office:

-Registro de Usuario.

Registrarse como nuevo usuario en la plataforma.

-Autenticación de usuario.

Los usuarios registrados tienen la posibilidad de loguearse para acceder a funcionalidades privilegiadas.

-Crear penca privada

Los usuarios registrados del sistema tienen la posibilidad de crear pencas privadas. pudiendo mandar un mail de invitación privado a posibles participantes de esa penca.

-Registrarse en una penca pública.

Los usuarios registrados podrán registrarse en una penca pública la cual tiene una modalidad de pozo compartido. Dichas pencas tendrán un costo de entrada para cada usuario que participe y los ganadores dividirán el pozo generado

Módulo Back office:

-Crear penca pública.

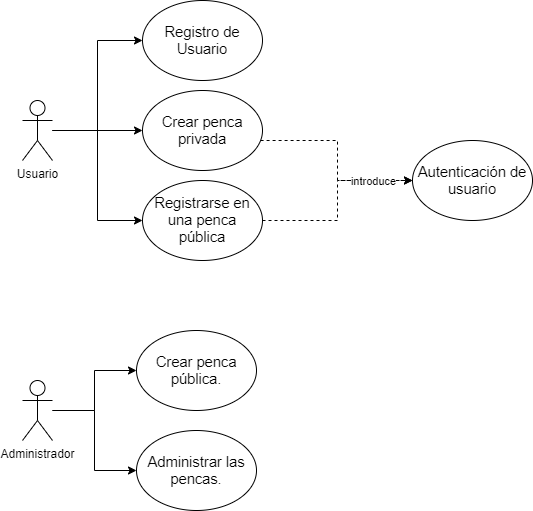
Consiste en dar de alta una nueva penca pública en el sistema.

-Administrar las pencas.

El administrador podrá ver y modificar los datos de las pencas según considere necesario. Por ejemplo agregar eventos o los resultados de los mismos.

## 3 Diagrama de Casos de Uso Críticos

| Caso de Uso | Secuencia de acciones con un fin específico |
| --- | --- |
| Actor | Persona, organización o sistema externo que interactúa con el sistema |
| Asociación | Relación entre un Caso de Uso y un Actor que participa en él. |
| Frontera del Sistema | La Frontera del Sistema se indica con un rectángulo que encierra los Casos de Uso. |



## 3.1 Actores

### 3.1.1 Usuario

Personas que están registradas en la plataforma.

### 3.1.2 Administrador

Son quienes crean, configuran y monitorean el sistema

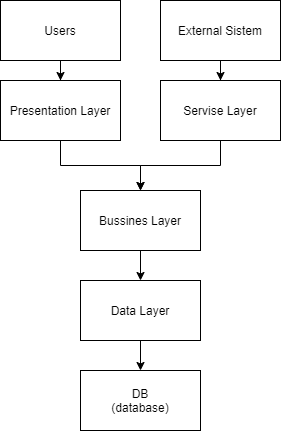
## **3.2** **Especificación de Casos de Uso Críticos**

| **Nombre**: Autenticar Usuario |
| --- |
| **Objetivo**: Autenticar al Usuario en el portal web. |
| **Actores**: Usuario. |
| **Precondiciones**: - |
| **Descripción:** El Usuario ingresa su nombre y password, el sistema valida los datos e ingresa al usuario al portal web. |
| **Flujo Normal**:  1. El Usuario ingresa su nombre y password.  2. El Sistema verifica que los datos ingresados sean correctos. |
| **Flujos Alternativos**:  2A. La combinación nombre de usuario/password es incorrecta.  1. El sistema despliega un mensaje de error acorde.  2. Vuelve a 1. |
| **Poscondiciones**: El usuario queda logueado en el sistema. |

**4**  **Vista Lógica**

**4.1**  **Estilo Arquitectónico.**

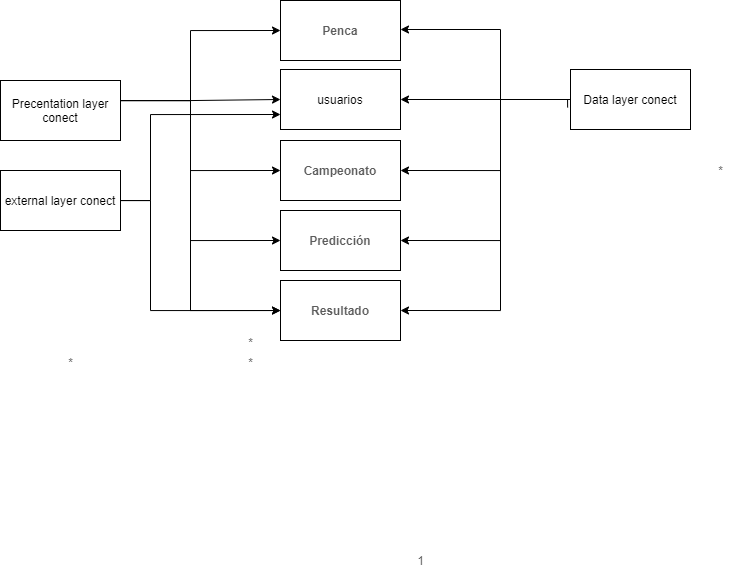
En el diseño arquitectónico de TuPenca se ha decidido utilizar un estilo arquitectónico en capas. En este esquema las capas más altas utilizan servicios definidos en las capas más bajas.

****

## **5** **Subsistemas**

### **5.0.1** **Business Layer.**

#### **5.0.1.1** **Diagrama de Componentes.**



#### **5.0.1.2** **Descripción.**

Implementa la funcionalidad principal de la aplicación TuPenca. Encapsula la lógica de negocio relevante para la aplicación. Consiste en componentes, los cuales exponen (en algunos casos) interfaces para que otros utilicen. Esta componente controla el acceso a los servicios de negocio desde otras capas, la publicación de los servicios de negocio mencionados e invocación de la capa de persistencia. Se distinguen cinco componentes principales en la capa de negocio:

Usuarios:

Aquí se desarrolla toda la lógica de negocio referente al manejo de usuarios en el sistema.

Penca:

Aquí se desarrolla toda la lógica de negocio referente al manejo de las pencas. Las cuales tienen dos tipos publicas que se en-foca en las funcionalidades del módulo BackOffice para el manejo de los administradores y privadas que se en-foca en las funcionalidades del módulo FrontOffice para el manejo de los usuarios.

Campeonato:

Aquí se desarrolla toda la lógica de negocio referente al manejo de campeonatos en el sistema.Se en-foca en las funcionalidades del módulo BackOffice para el manejo de los administradores

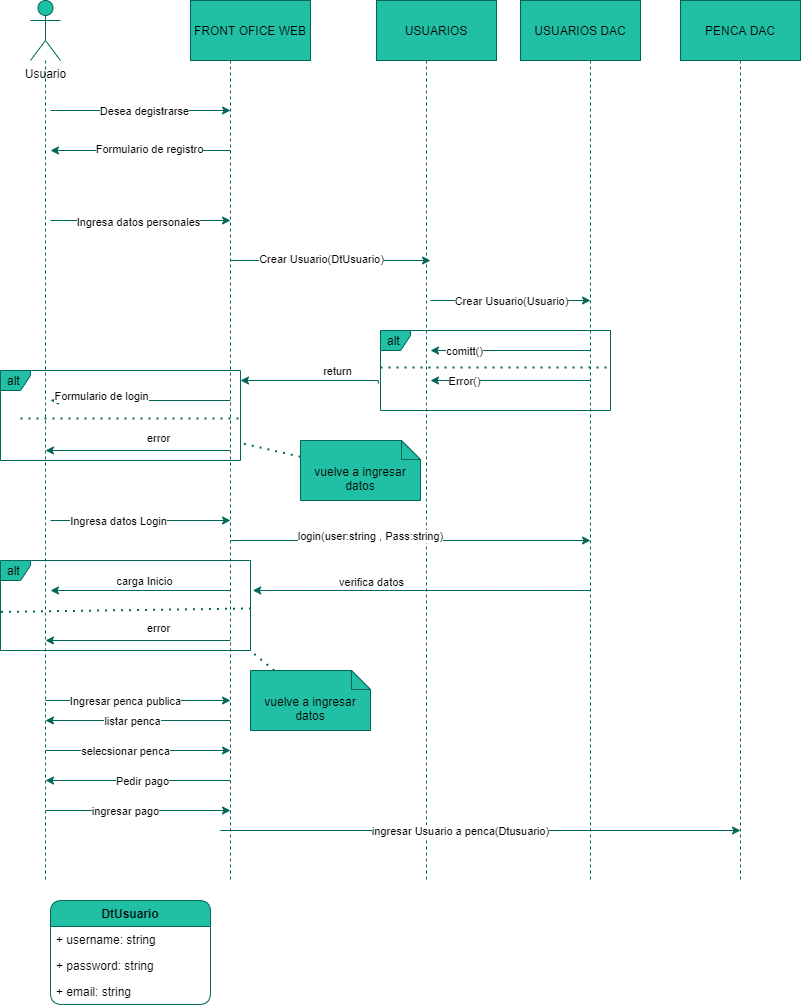
Predicción

Aquí se desarrolla toda la lógica de negocio referente al manejo de predicaciones en el sistema.Se en-foca en las funcionalidades del módulo FrontOffice para el manejo de los administradores

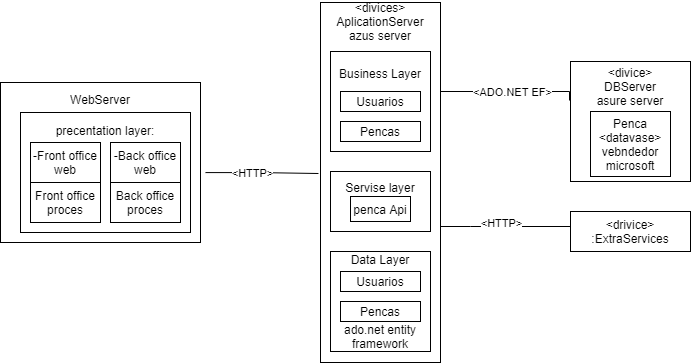
Resultado

Aquí se desarrolla toda la lógica de negocio referente al manejo de resultados en el sistema.Se en-foca en las funcionalidades del módulo BackOffice para el manejo de los administradores

**5.1**  **Diagramas de Interacción**



# **6** **Vista de Distribución (Deployment).**



**6.1** **Escenario 1**

**6.1.1**  **Descripción**

[Descripción general del escenario.]

### **6.1.2** **Nodos**

WebServer: Es lo que corre en el browser del navegador ya que es una aplicación spa

Aplicación: server: es donde va a estar la business Layer .

DBServer: es donde va a correr la base de datos

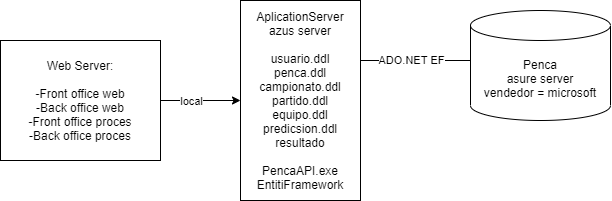
Extra Services: consumiría API de terceros

### **6.1.3** **Conexiones**

HTTP: consumir una api de terceros

ADO.NET EF: es la manera de comunicarse de .net con la base de datos

# **7** **Vista de Implementación.**



**8**  **Referencias**

[1] Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives. N. Rozanski, E. Woods. Addison-Wesley, 2005

[2] Software Architecture in practice, Second Edition. L. Bass, P. Clemens, R. Kazman. Addison-Wesley, 2003

[3] An Introduction to Software Architecture. D. Garlan, M. Shaw. 1994.<http://www.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf>

[4] Architectural Blueprints — The “4+1” View Model of Software Architecture. Kruchten, Philippe. 1995.<http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>

[5] Software Architecture Links. Bredemeyer Consulting.<http://www.bredemeyer.com/links.htm>

[6] Grupo 6, TSI1 edición 2012. Documento de Arquitectura. 2012.