**Nombre del Proyecto**

Gestión de Convocatorias

**Subasta Nro:**

**2017/000065**

**Orden de Compra Nro:**

**Empresa:**

**EXET S.R.L**

**Fecha de presentación:**

**15/11/2017**

**Numero de Entregable según pliego:**

**#1 RELEVAMIENTO**

## Ficha del documento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Revisión | Autor | Modificaciones | Verificado |
| 15/11/2017 | 1 | Alexis Massón | Versión inicial |  |

## Alcance del documento

El presente documento, si bien está enfocado en la finalización de Fase 1, la cual tuvo lugar durante los Sprints 1 y 2 (finalizando aproximadamente el 01/09/2017), incluye información actualizada al 27/10/2017 sobre los requerimientos y alcances del proyecto.

Contenido

[Ficha del documento 2](#_Toc498090843)

[Alcance del documento 2](#_Toc498090844)

[DATOS GENERALES 5](#_Toc498090845)

[Resumen Ejecutivo 5](#_Toc498090846)

[Roles en el Proyecto 7](#_Toc498090847)

[RELEVAMIENTO DETALLADO 8](#_Toc498090848)

[Estado de situación actual 8](#_Toc498090849)

[Supuestos 8](#_Toc498090850)

[Organizacionales – Funcionales 8](#_Toc498090851)

[Tecnológicos 8](#_Toc498090852)

[Restricciones 9](#_Toc498090853)

[ESPECIFICACIONES DE REQUERIMIENTOS SEGÚN IEEE 830 10](#_Toc498090854)

[Introducción 10](#_Toc498090855)

[Descripción General 10](#_Toc498090856)

[Suposiciones y dependencias 12](#_Toc498090857)

[Análisis y Diseño Detallado 14](#_Toc498090858)

[Documento de Arquitectura (diseño lógico/técnico de la aplicación/relación con otros sistemas) 14](#_Toc498090859)

[Descripción de arquitectura 14](#_Toc498090860)

[Capas de arquitectura 16](#_Toc498090861)

[Objetos de transferencia de datos y capas de arquitectura 16](#_Toc498090862)

[Paquetes de software 18](#_Toc498090863)

[Diagrama de Estado 20](#_Toc498090864)

[Estados del expediente 20](#_Toc498090865)

[Estados del consultor 21](#_Toc498090866)

[Diagrama de base de datos (DER, formato Erwin) 1](#_Toc498090867)

[Prototipos de interfaz 1](#_Toc498090868)

[ANEXOS 1](#_Toc498090869)

## DATOS GENERALES

### Resumen Ejecutivo

**Objetivos del proyecto**: Diseñar e implementar un sistema de información web para: la Gestión de Concursos del Polo Audiovisual Córdoba con sus Inscripciones; el seguimiento/control de cumplimiento documental de los presentantes que tengan la voluntad de inscribirse a dichos concursos; el Registro de Jurados, e interacción con las convocatorias asignadas; el Registro y validación de Artistas y Técnicos; la interacción de los tutores con proyectos ganadores.

El software debe contemplar que se pueda registrar, consultar y validar el cumplimiento de la documentación necesaria para que los presentantes inscriptos, sean evaluados por los jurados y una vez que sean acreditados como ganadores, puedan acceder a los subsidios propios del fondo de fomento de acuerdo a la Ley 10.381.

**Motivos del proyecto**:

La ley de fomento Nro. 10.381 tiene por objetivo la creación de un Fondo para fortalecer la producción audiovisual tanto en su dimensión cultural como en la industrial en todo el territorio provincial.

La gestión de inscripciones tiene por objetivo la evaluación del grado de cumplimiento documental por parte de los presentantes para así hacer uso de los beneficios del fondo de fomento.

Por otro lado, se detecta la posibilidad de mejorar sustancialmente los actuales procesos de negocio que desarrolla la Agencia Córdoba Cultura, mayormente basados en registros manuales. El uso de nuevas tecnologías de información y la digitalización de procesos generará un salto de calidad, favoreciendo las operaciones diarias que desempeñan funcionarios y colaboradores; brindando además información relevante para la toma de decisiones.

**Que se espera del proyecto:**

Con la implementación de este proyecto se espera:

* Sistematizar el procedimiento de Inscripciones a los Concursos del Polo Audiovisual Córdoba.
* Garantizar al ciudadano el acceso público a la lista de Artistas y Técnicos aprobados.
* Facilitar el trabajo del personal de la Agencia Córdoba Cultura.
* Incrementar la eficiencia y la productividad organizacional.
* Asegurar en lo que hace al alcance del software el cumplimiento de los procedimientos internos.
* Mejorar los tiempos de acceso para la búsqueda de documentos.
* Proteger la privacidad de información documental incorporándola al centro de documentación.
* Ahorrar espacio para el archivo físico de los trámites.
* Evitar el desgaste o pérdida de los documentos originales.
* Actualización y modernización tecnológica de trámites y procedimientos del área
* Incrementar la eficiencia y la productividad organizacional.
* Reducir el uso de papel
* Mejorar el control en el cumplimiento de documentación requerida para el seguimiento de las inscripciones a las Convocatorias.
* Mejorar el control en el cumplimiento de documentación requerida para el registro de Artistas y Técnicos
* Brindar información útil para la gestión, operación diaria y toma de decisiones en todos los niveles organizacionales

**Duración del proyecto**:

Duración del proyecto: Se contratará un total de dos mil ochocientos ochenta (1440) horas de prestación de servicio a brindar en un plazo máximo de tres (3) meses.

### Roles en el Proyecto

* Responsable: Guillermo Ferreyra
* Sponsor: Ministro de industria…?
* Responsable Funcional: Cecilia Hernández
* Empresa de desarrollo: Exet
* Ejecución del Proyecto: Ing. Alexis Massón

## RELEVAMIENTO DETALLADO

### Estado de situación actual

El presente documento resume los hitos logrados en la primera fase del proyecto, dedicada a la Ingeniería de Requisitos.

Los resultados logrados fueron un completo relevamiento de las necesidades del área tras el análisis organizacional y de procesos que ejecuta la Agencia Córdoba Cultura. Además, en paralelo se ejecutaron tareas de gestión de proyecto y de definición de metodología de trabajo para organizar el equipo que llevaría adelante las tareas técnicas de programación, despliegue y testing del software.

Los requisitos identificados son la entrada a la fase de análisis y diseño en la cual se da forma al software resultante del proyecto y a un conjunto de artefactos que acompañan y guían el proceso de desarrollo de software.

### Supuestos

### Organizacionales – Funcionales

Se da por supuesto que:

* se dispondrá de toda la información relativa a los procesos de convocatorias en todas las áreas involucradas.
* todos los actores comprenden el alcance del proyecto
* todos los actores están en un todo de acuerdo con la implementación y van a promover el uso obligatorio del nuevo sistema.
* se dispondrá de las herramientas de contratación del equipo de trabajo.
* los empleados públicos y funcionarios que sean considerados “stakeholders” para el proceso de elicitación y relevamiento de requisitos de software prestarán colaboración y asistirán a las reuniones debidamente calendarizadas y coordinadas, facilitando material y documentación necesaria por los especialistas del equipo de trabajo

### Tecnológicos

Se da por supuesto que:

* se contará con la infraestructura de supercentro como servidor de aplicación, servidor de bases de datos y servidor de desarrollo.
* las áreas técnicas de Gobierno (base de datos, infraestructura, comunicaciones, operaciones, así como también los referentes de los sistemas informáticos necesarios para la integración de librerías e interfaces de servicios a utilizar por el software a desarrollar) darán respuesta en tiempos razonables a fin de no comprometer los objetivos del proyecto y los plazos planificados
* se contará con las licencias de software de las herramientas seleccionadas y serán provistas por el Gobierno Provincial.
* todos los usuarios de la herramienta de gestión informática contarán con el equipamiento informático adecuado para su ejecución.
* el equipo de trabajo contará con las herramientas informáticas (hardware y software) necesarios.

### Restricciones

* Se da por supuesto que el desarrollo deberá cumplir con los Estándares informáticos establecidos por el Gobierno Provincial

## ESPECIFICACIONES DE REQUERIMIENTOS SEGÚN IEEE 830

### Introducción

#### Propósito

Esta sección del documento es una Especificación de Requisitos Software (ERS) para el Sistema de información de gestión de convocatorias para el Ministerio de Industria, Comercio y Minería de la Provincia de Córdoba. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.

### Descripción General

#### Perspectiva del producto

El producto es autónomo e independiente de otros. Únicamente tiene dependencias con sistemas / módulos de sistemas de gobierno de los cuales consume servicios o entrega información, como por ejemplo SUAC para el seguimiento de expedientes y notas.

#### Funciones del producto

El producto a desarrollar estará compuesto por los siguientes módulos de software:

* **Configuración**: involucra los mecanismos necesarios para la parametrización y adaptación del software, tales como la gestión de jurados, bloques, géneros y categorías de las convocatorias y checklist de cumplimiento de documentación.
* **Seguridad**: incluye aspectos como la gestión de roles de usuario y sus correspondientes permisos de acceso y la asignación de roles a usuarios del sistema. La autenticación al sistema es canalizada a través de CIDI como mecanismo unificado de inicio de sesión para aplicaciones del Gobierno de la Provincia de Córdoba. La validación del presentante cumpla con el Nivel 2 y residencia en Córdoba para su registro.
* **Gestión de inscripciones**: permite consultar la lista de inscripciones con la información pertinente a los datos de los presentantes con su documentación, para así gestionar el estado de dicha inscripción y el estado de los documentos solicitados.
* **Gestión de presentantes**: módulo de software que permite consultar y analizar información relacionada a los presentantes.
* **Últimos eventos:** permite un rápido acceso y visualización de las convocatorias abiertas y a los artistas y técnicos que quedan por validar su documentación.
* **Gestión de Presentates**: permite la consulta de los presentantes con sus datos que han sido registrados en el sistema.
* **Notificación automática**: módulo que el cumplimiento de ciertos ítems, generando notificaciones automáticas previas al vencimiento del periodo admisible para satisfacer la obligación. Las notificaciones son enviadas a las partes involucradas en función del ítem que se tratase, tales como el vencimiento, rechazo o aprobación de la documentación con necesiten la información brindada.
* **Portal del Presentante:**  módulo encargado del registro del Presentante al sistema, donde podrá realizar la inscripción a una convocatoria brindando documentación solicitada, consultar y gestionar inscripciones ya realizadas, así también pudiendo ver el estado en que se encuentran y el jurado que está a cargo de la convocatoria manteniendo la trasparencia. Seguimiento de las cuotas para una vez ganado la convocatoria.
* **Portal del Tutor:** módulo en el que Tutor podrá consultar y gestionar los proyectos ganadores de la convocatoria asignada, de manera que sea capaz de aprobar o rechazar el cumplimiento de documentación brindada por los ganadores para cada etapa según la cuota correspondiente.
* **Gestión de Técnicos y Artistas**: es una región pública del sistema que brinda al ciudadano acceso a información de Técnicos y Artistas que han sido aprobados por el Administrador. Este último, y sólo este, tendrá la responsabilidad de aprobar o rechazar los Técnicos y Artistas que han sido postulados para tal cargo, según la documentación que presentaron.

#### Características de los usuarios

Identificamos los siguientes actores de sistema de información:

* **Ciudadano:** es el perfil del público en general que podrá acceder para consultar la lista de artistas y técnicos que han sido aprobados.
* **Administrador**: es un perfil con un trasfondo técnico que conozca sobre la administración de usuarios y características de configuración del sistema a fin de poder parametrizarlo y adaptarlo a las necesidades del equipo del Polo. Tiene la posibilidad de administrar las convocatorias e inscripciones, por tanto debe quedar a cargo de una persona responsable y apta para la tarea.
* **Jurado:** es un perfil especializado en la revisión y selección de ganadores a las convocatorias que han sido asignados, teniendo acceso a toda documentación publicada de los presentantes inscriptos.
* **Tutor:** es un perfil especializado en el seguimiento y control de la documentación de los proyectos ganadores que tienen asignados, validando que la documentación sea la correspondiente según cada cuota.
* **Presentante:** es un perfil asignado a cualquier persona que desee registrarse en el sistema que cumpla con un nivel 2 de CIDI y una residencia en Córdoba, que tenga la voluntad de inscribirse a las convocatorias.
* **Artista / Técnico:** pueden acceder al registro y así promocionarse en la Lista de Artistas y Técnicos dentro del sistema. No pueden realizar otras acciones.

#### Restricciones

* El software a desarrollar debe implementarse con el lenguaje de programación Microsoft C# utilizando el Framework .NET y conectado a base de datos Oracle 11g provista por el Gobierno de la Provincia de Córdoba

### Suposiciones y dependencias

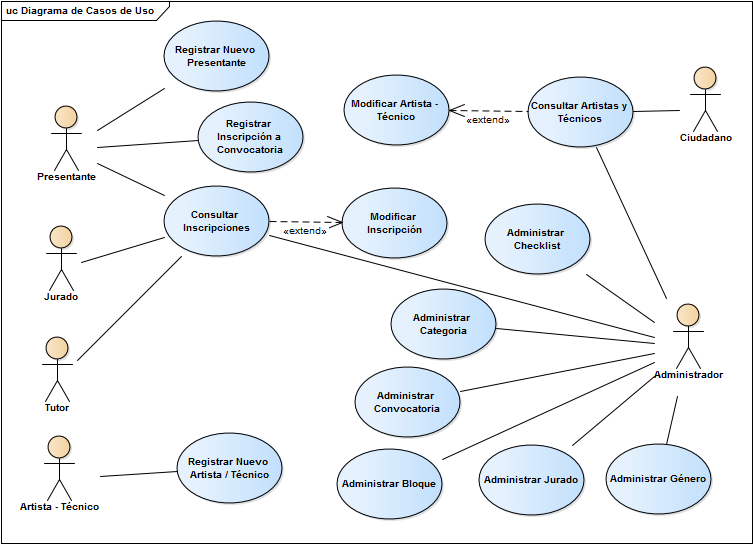
#### Integraciones / dependencia de sistemas externos

El software a desarrollar por cumplir con estándares de Gobierno aprovecha y depende de sistemas verticales que brindan servicios de información detallados a continuación:

* **SUAC**: Creación y consulta de notas / expedientes: es el sistema único de atención del ciudadano que gestiona expedientes y notas de carácter legal. Nuestro sistema depende de SUAC para la emisión y seguimiento de expedientes y notas.
* **CIDI**: Sistema de usuarios, gestión de documentos digitales y comunicaciones. CIDI centraliza la autenticación de usuario en los aplicativos alojados en Red de Gobierno de la Provincia de Córdoba. La autorización es personalizada y propia de nuestra aplicación y NO depende de CIDI.
* **PACK Comunicaciones:** Es un componente de base de datos que centraliza la información de e-mails, teléfonos y datos de contacto asociados a personas físicas y/o jurídicas. Nuestra aplicación necesita registrar y actualizar información de esta naturaleza y lo hace a través de este PACK de vistas y procedimientos almacenados de Oracle.

### Requisitos específicos

Se define a continuación el listado de requisitos identificados, organizados por actor de sistema de información:



#### Restricciones de diseño

* Se debe emplear el Front-end Framework “Twitter Bootstrap” y como referencia el Theme “Metronic”.
* La aplicación debe correr sobre servidor IIS v7+

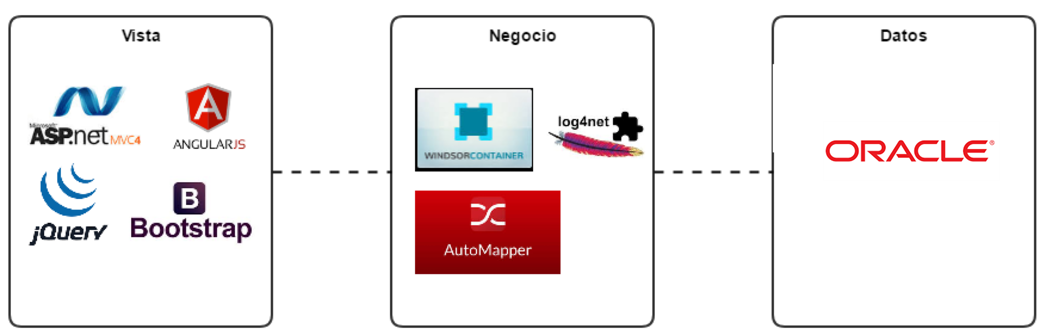
## Análisis y Diseño Detallado

### Documento de Arquitectura (diseño lógico/técnico de la aplicación/relación con otros sistemas)

### 

### Descripción de arquitectura

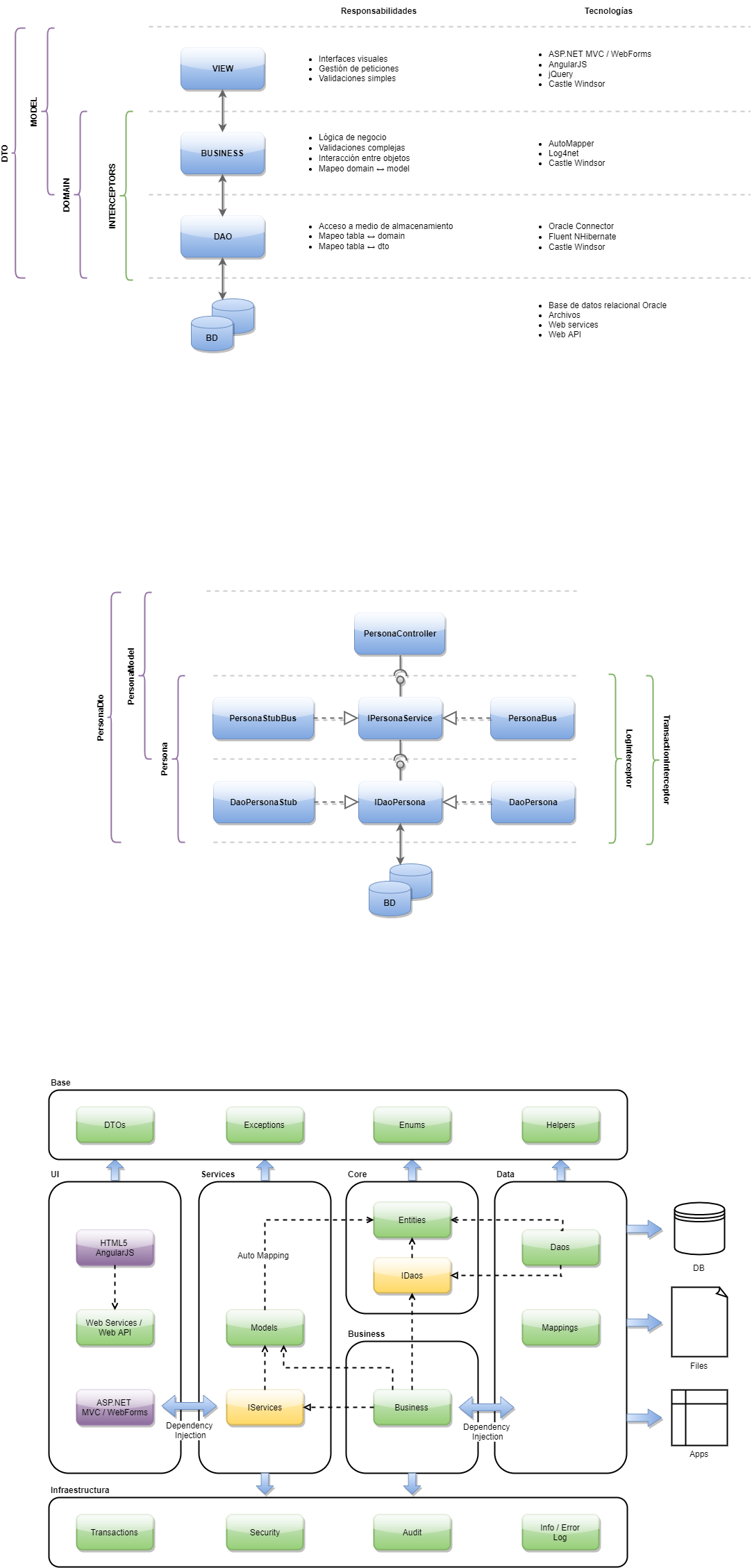
La versión actual de la plataforma está desarrollada con una arquitectura .NET con una clara separación en capas, que incluye frameworks avanzados para garantizar mantenibilidad y permitir un rápido desarrollo / evolución antes cambios que detallamos a continuación:



* Vista: la capa visual / front-end está basada en la plataforma ASP .NET MVC 4 en lo que respecto a la aplicación web y ASP .NET Web API para los servicios web utilizados por el aplicativo móvil.  
    
  Se utiliza Bootstrap 3 como front-end framework para facilitar el maquetado y darle una estética profesional a la aplicación web.   
    
  En general el sistema utiliza JQuery como librería de alto nivel de Javascript. Pero en pantallas como el punto de venta y la gestión de compras, donde se requiere gran dinamismo e interactividad entre la vista y el servidor se emplea Angular JS.
* Negocio: la capa de reglas de negocio / lógica utiliza Windsor Container como framework para la inyección de dependencia (sin embargo, esta característica no solo está disponible en esta capa, sino también en la capa “vista”).   
    
  La inyección de dependencia permite incorporar programación orientada a aspectos, lo que facilita introducir comportamiento transversal a la aplicación como por ejemplo el logueo automático de errores, la gestión automática de transacciones de base de datos y la auditoría a nivel usuario de operaciones efectuadas en el sistema.   
    
  Utilizamos Automapper como librería para la conversión automática entre entidades de dominio y models de la capa visual.   
    
  Utilizamos “Log4Net” como librería para el logueo automático de errores, la cual se inyecta como un “aspecto” gracias a la inyección de dependencia que permite WindsorContainer.
* Datos: la capa de datos se basa en un “DAO” (data access objects”) que se comunica con un motor Oracle v11g. La lógica de base de datos en general es encapsulada en procedimientos almacenados según requieren los estándares de gobierno de la Provincia de Córdoba.
* Test / pruebas: tenemos sets de Test Unitarios para los módulos críticos, especialmente aquellos que manejan dinero o realizan operaciones contables. Utilizamos NUnit como framework de test unitario.

### Capas de arquitectura

Detallamos a continuación las capas que conforman la arquitectura, sus responsabilidades y tecnologías involucradas.



### 

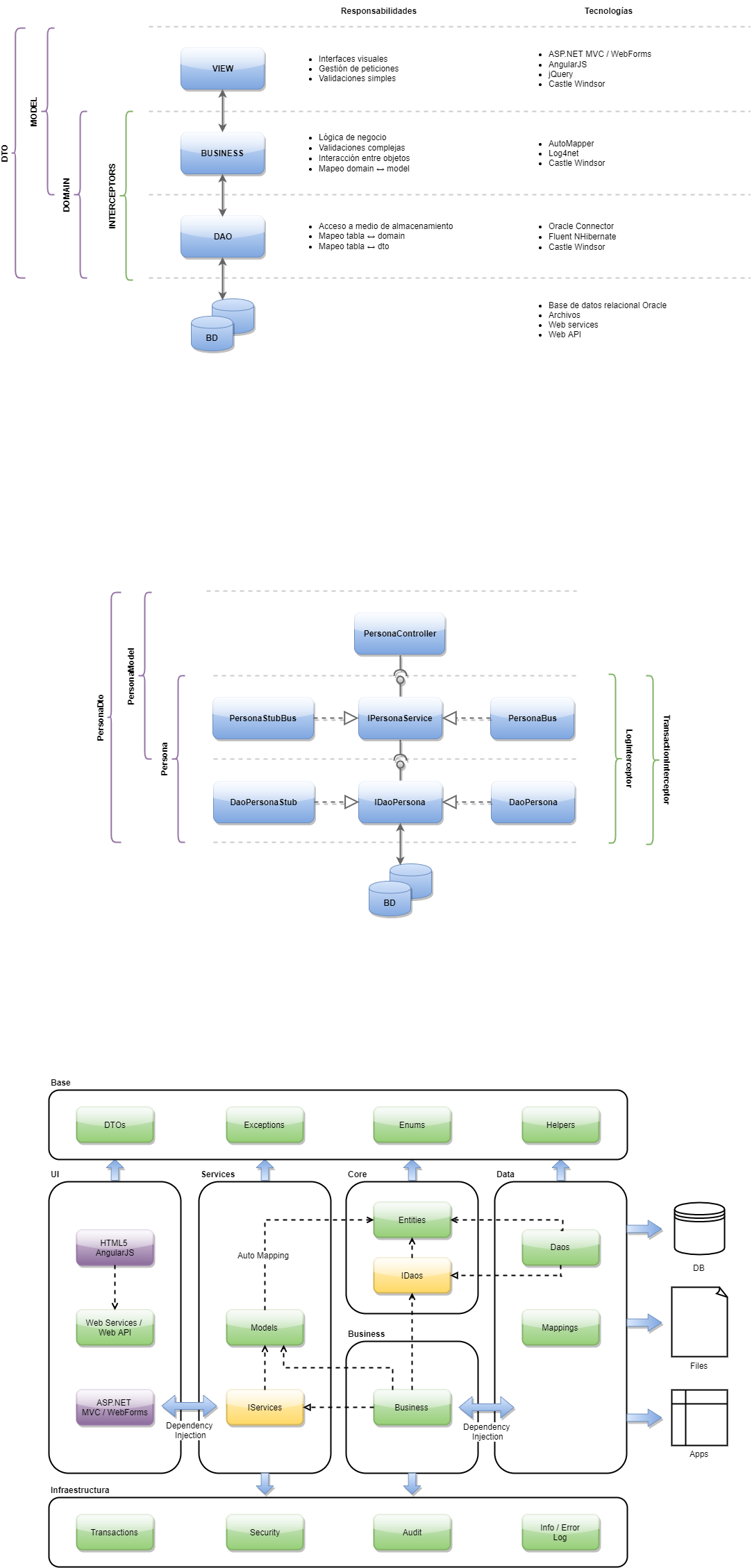
### Objetos de transferencia de datos y capas de arquitectura

Definimos distintos objetos que mantienen la comunicación entre las distintas capas de arquitectura:

* DTO: Data transfer object. Objeto de transferencia de datos general. Suele utilizarse para la ejecución de consultas de datos que mapean contra varias tablas o consultas multitabla en la base de datos relacional y puede viajar a través de todas las capas de arquitectura.
* Model: Modelo. Mapea una vista de UI. Permite la comunicación entre la capa visual y la capa de negocio.
* Domain: Representa un objeto de dominio del mundo real. Se utiliza en la capa de negocio para realizar validaciones. Es mapeado a tabla relacional en la capa de datos. No existe en la capa visual.
* Interceptores. Definen comportamiento transversal a la aplicación a través de mecanismos de inyección de dependencia.

#### Ejemplo de implementación concreta de componentes de arquitectura

Se ejemplifica la arquitectura descrita utilizando como ejemplo el caso del objeto de dominio “Persona”. Se materializan las representaciones de dicho objeto en cada componente de arquitectura de software.

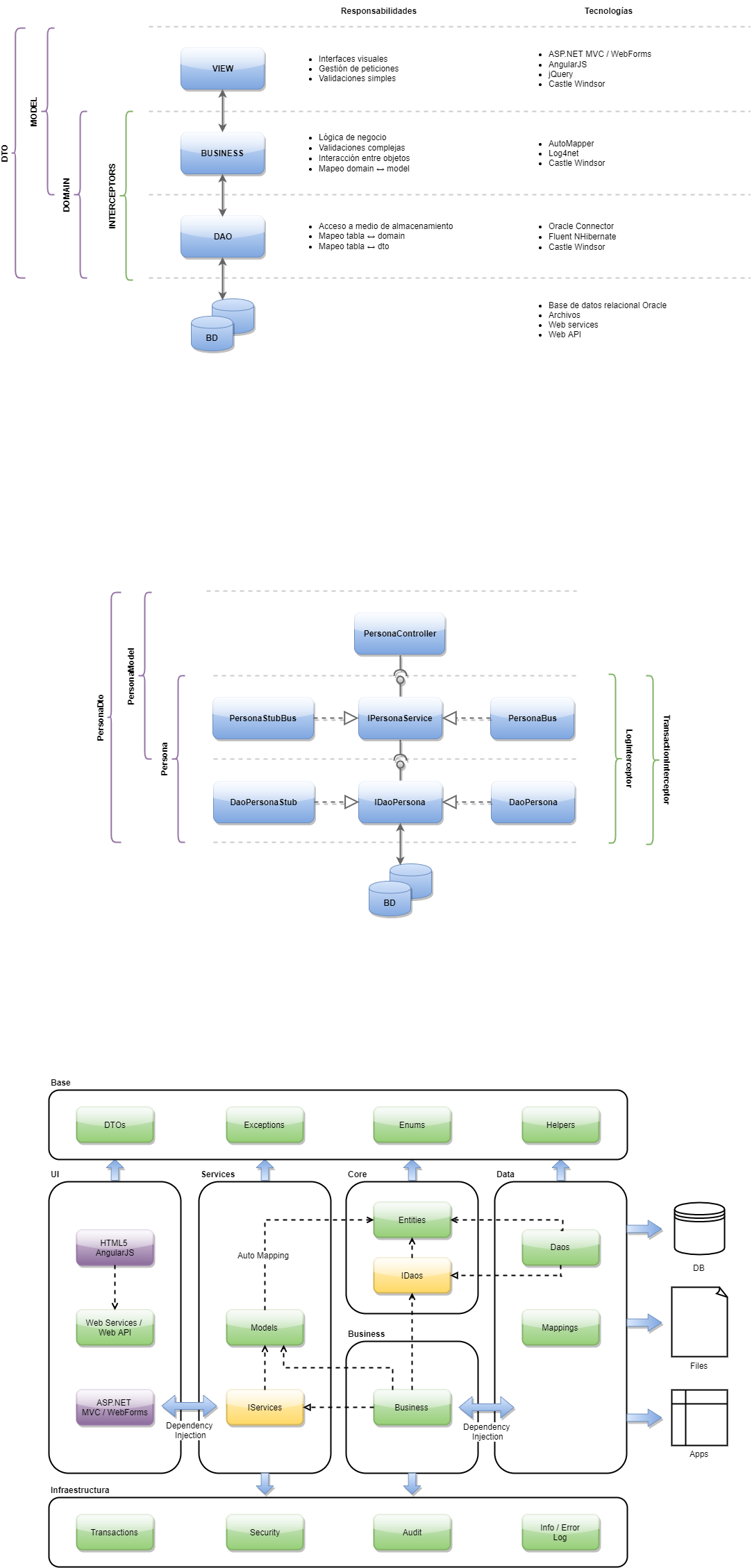


Como se visualiza en el ejemplo, se definen interfaces de objetos para mantener la dependencia entre las capas:

* IPersonaSevice: define una interfaz de objetos de alto nivel del contrato que cumple la capa de negocio y los servicios brindados a la capa de visual.
* IDaoPersona: define una interfaz de objetos de alto nivel del contrato que cumple la capa de datos y los servicios brindados a la capa de negocio. Para cada interfaz de objetos se puede satisfacer con una implementación real o una implementación “fake” o falsa, que nos permite mantener isolada a ciertas zonas de la arquitectura de las dependencias externas que no estén disponibles. Esto permite avanzar en el desarrollo de ciertos módulos de software independientemente de la completitud de programación de librerías de servicios externos que eventualmente pueden estar incompletas, no funcionales, o simplemente no implementadas.

### Paquetes de software

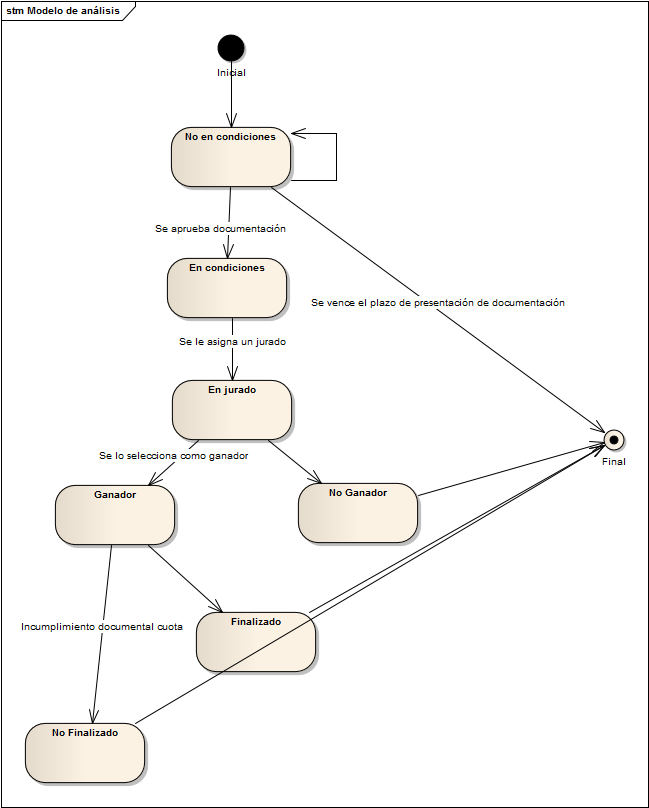
Organizamos los componentes arquitectónicos en paquetes funcionales que se materializan generalmente en librerías de clases .NET y/o proyectos ejecutables (Web/Unit Test).



* Base: define artefactos de software globales a la arquitectura y compartidos por diferentes capas
  + DTOs: objetos para transferencias de datos
  + Exceptions: excepciones personalizadas de la arquitectura
  + Enums: enumeraciones / constantes numeradas generales a la arquitectura
  + Helpers: librerías útiles de aplicación general
* UI: proyecto de interfaz gráfica de usuario (Aplicación Web .NET MVC 4)
* Services:
  + Models: objetos de mapeo de UI a capa de negocio
  + IServices: interfaces de servicio necesarias por capa visual
* Core: define entidades de negocio / dominio
  + Entities: clases que representan objetos de dominio. Tiene correspondencia con diagrama de clases
  + IDaos: Interfaces de servicio para materialización/desmaterialización de entidades de dominio.
* Data: capa de datos
  + Daos: artefactos para mapeo de objeto/relacional. Se encargan de la persistencia de entidades de dominio.
  + Mappings: archivos de mapeo, definen excepciones de mapeo de objeto/relacional.
* Infraestructura: define mecanismos transversales a la aplicación a través de “Programación orientada a aspectos” que permite la inclusión de Castle Windsor como Framework de *Inyección de dependencia.* Los siguientes mecanismos son introducidos como interceptores / aspectos:
  + Transactions: administran la gestión de transacciones que se originan en la comunicación entre capa de negocio y capa de datos. Definen puntos del salvado automático y rollback ante la ocurrencia de excepciones de aplicación.
  + Security: define esquema de seguridad, autenticación y autorización.
  + Audit: define mecanismo de auditoría y registro de actividad
  + Info / Error log: define mecanismo de logueo automático de información de uso y errores ocurridos en la ejecución de la aplicación.

## Diagrama de Estado

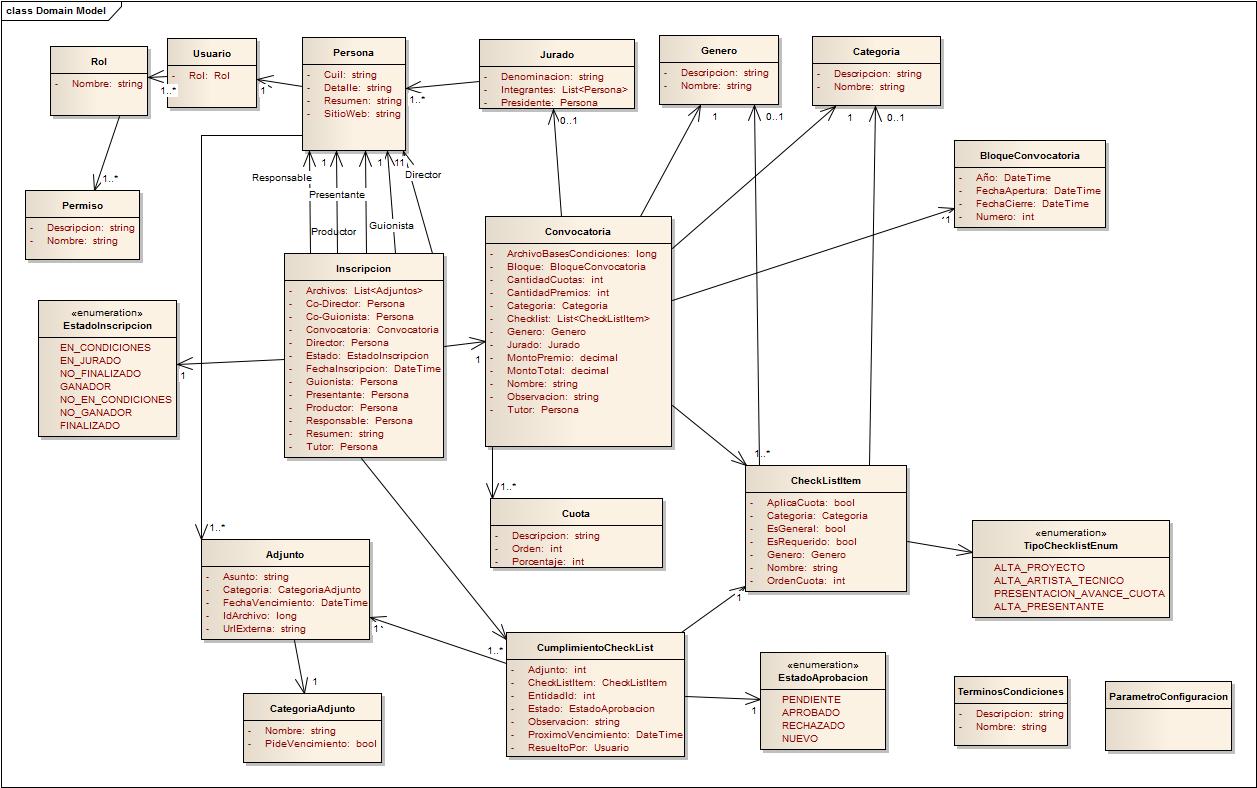
### Estados de la Inscripción



|  |  |  |  |
| --- | --- | --- | --- |
| **Estado inicial** | **Estado final** | **Condición de guarda** | **Efecto** |
| No en condiciones | En Condiciones | Se aprueba la documentación que presentó para la convocatoria |  |
| No en Condiciones | No en Condiciones | Faltante en documentación solicitada | Se envía notificación al presentante indicando la documentación faltante |
| En Condiciones | En Jurado | Se le asigna un jurado |  |
| En Jurado | Ganador | El jurado selecciona la inscripción como ganadora | Se le envía una notificación al presentante indicando que ha sido ganador |
| En Jurado | No Ganador |  |  |
| Ganador | Finalizado | El presentante cumple con toda la documentación para cada cuota |  |
| Ganador | No Finalizado | El presentante debe presentar la documentación solicitada para la cuota |  |

## Diagrama de clases

Se incluye a continuación el diagrama de clases que se utilizará para modelar en Oracle el esquema de base de datos utilizado por la aplicación a desarrollar.



## Prototipos de interfaz

Los prototipos de interfaz gráfica son presentados en el “Anexo I – Prototipos de UI”.

La descripción de dichos prototipos se incluye en el “Anexo II – Descripción de prototipos de UI”

## ANEXOS

Se incluyen los siguientes anexos que forman parte del presente documento:

* Anexo I: Prototipos de UI

https://docs.google.com/document/pub?id=1VmuKwSvXPf8XEbwatLhP9eWdyCis2xPo4PWQgCwayH0