



## PROYECTO FINAL DE SEGUNDO AÑO

### CARRERA DE ANALISTA DE SISTEMAS ANALISTA PROGRAMADOR WEB .NET 2019

#### Generalidades

- La entrega final se hará durante la última clase de tutoría. Durante las clases de tutorías deberán realizarse entregas parciales de acuerdo a lo establecido en la sección Tutorías & Entregas.
- Se deberá realizar la entrega mediante correo electrónico, del software junto con toda la documentación exigida en la sección **Requerimientos de Entrega**:
  - **Asunto:** Entrega Proyecto Final ANP 2018
  - **Destinatarios:** segundoanalista@bios.edu.uy y tutor asignado (OBLIGATORIO)
  - **Compartido por Drive:** documentación y solución completa
  - **Cuerpo:** Nombre y Cedula de los integrantes del grupo que realiza la entrega
- Se deberán formar grupos de 1 a 3 personas, **los cuales deberán inscribirse en bedelía desde el día 2 de diciembre hasta el día 12 de diciembre. No se aceptaran inscripciones fuera de fecha.** Luego de esto, Bedelía **publicara** la asignación de tutores así como de horarios. **La inscripción es únicamente por vía mail, con el siguiente formato:**
  1. Asunto: *Inscripción a Proyecto Final Segundo Año*
  2. Contenido:
    - a. Cedula y Nombre de **todos** los integrantes del grupo que se presenta
    - b. Franja Horaria a la que se concurre a clase
- La asistencia a la última tutoría es obligatoria para **todos** los miembros del grupo ya que se realizara la defensa en máquina del proyecto.

#### Idea General

Se desea unificar, mediante el desarrollo de un sitio web, la solicitud de tramites online para el departamento de Montevideo

La página principal del sitio web desplegara los tipos de trámites actualmente disponibles para realizar solicitudes. Además, dado un número de solicitud, se podrá consultar el estado de la misma.

Todos los usuarios del sistema deberán tener número de documento (**único** en el Sistema), contraseña (3 caracteres + 2 números + 1 símbolo), y nombre completo. De los usuarios de tipo Solicitante, también se deberá saber teléfono de contacto. De los usuarios de tipo empleado, se deberá saber el horario de trabajo (compuesto por la hora de inicio y de fin de tareas).

#### Arquitectura Solicitada

- Los usuarios Visitantes / Solicitantes podrán acceder a la información a través de un sitio web, que será publicado en un servidor contratado para dicha funcionalidad.
- Los usuarios empleados tendrán acceso a realizar sus tareas a través de una aplicación de escritorio.
- La lógica de negocio del sistema estará ubicada dentro de un servidor de aplicaciones, y se podrá acceder a ella mediante un contrato de servicio publicado en el mismo servidor.
- La base de datos estará instalada en un servidor de datos. La persistencia se comunicara con dicho servidor mediante ADO.NET.



## Funcionalidades Mínimas de la Aplicación de Escritorio

**Formulario Windows:** Logueo

**Actor Participante:** Empleado

**Resumen:** Este formulario permite que un usuario se loguee y acceda al formulario principal de esta aplicación. Para ello se deberá solicitar cedula y contraseña. **Tomar en cuenta, que todos los accesos a Base de Datos luego de loguearse, deberán realizarse con el usuario SQL asociado al empleado logueado sin excepción alguna.** Cuando el usuario se loguea, se deberá crear un archivo con formato XML, en el mismo directorio de la aplicación, con la siguiente información: usuario logueado, horario inicial de tareas y horario de finalización de tareas. Cuando el usuario se desloguea, el archivo deberá eliminarse.

**Formulario Windows:** ABM de usuarios tipo Empleado

**Actor Participante:** Empleado

**Resumen:** Este formulario permite realizar alta, baja y modificación de usuarios tipo empleados. Tomar en cuenta los datos necesarios que se especificaron en la sección “**Idea General**”. Además, se deberá crear en el Servidor SQL, un usuario de logueo y un usuario de base de datos, asociado a dicho empleado. Estos serán utilizados cada vez que el usuario quiera acceder a la base de datos. Tomar en cuenta la seguridad y acciones que debe realizar un usuario tipo empleado, para la determinación de permisos. Considerar que solo el propio usuario podrá cambiar la contraseña.

**Formulario Windows:** ABM de Documentación (solicitada para tramites)

**Actor Participante:** Empleado

**Resumen:** Este formulario permite realizar alta, baja y modificación de documentaciones solicitadas para realizar trámites (cedula, credencial, partida de nacimiento, etc.). Se deberá saber: Código interno (identificador único en el sistema, y **no** autogenerado), nombre, y donde se puede conseguir (por ejemplo: la partida de nacimiento se solicita en el Registro Civil). Tomar en cuenta que eliminar un tipo de documentación del sistema, **no implica** eliminar los tipos de tramites asociados.

**Formulario Windows:** ABM de Tipos de Tramite

**Actor Participante:** Empleado

**Resumen:** Este formulario permite realizar alta, baja y modificación de tipos de trámites disponibles. Se deberá saber: Código (identificador único en el sistema, compuesto por 4 números que representan el año, y 5 letras), nombre, descripción, precio y el conjunto de documentación exigida para realizar el trámite (solo puede seleccionarse de la ingresada en el sistema). Tomar en cuenta que eliminar un tipo de trámite del sistema, **no implica** eliminar las solicitudes asociadas.

**Formulario Windows:** Cambio Estado Solicitud

**Actor Participante:** Empleado

**Resumen:** Este formulario permite cambiar el estado actual de una solicitud. Se deberá desplegar en una grilla, todas las solicitudes que estén en estado **alta**. El usuario podrá seleccionar una solicitud, y de forma automática se le asignara un estado, en función de la columna utilizada para la selección (una columna para anular y otra para definir que se ejecutó el trámite). Los estados son: **Alta** (automático cuando se realiza la solicitud), **Ejecutada** (cuando se realizó el tramite), y **Anulada** (cuando se desea eliminar la solicitud de tramite). Tomar en cuenta, que el usuario **NO** elige el estado para el cambio, se hace automáticamente. Realizada dicha acción, la lista deberá actualizarse.



---

**Formulario Windows:** Listado de Solicitudes

**Actor Participante:** Empleado

**Resumen:** Este formulario mostrará por defecto la lista completa de solicitudes *del año en curso*. Deberá poderse filtrar dicha lista por:

- A. **Resumen Tipo de Trámite:** despliega para cada tipo de trámite, nombre y cantidad total de solicitudes realizadas (sin importar el estado), ordenados por nombre de trámite.
- B. **Resumen Mensual:** despliega para cada mes del año actual, mes y cantidad de solicitudes
- C. **Resumen uso de Documentación:** despliega para cada tipo de documentación, la cantidad de solicitudes que la exigen.
- D. **Filtro por Fecha:** despliega todas las solicitudes de una fecha específica, ordenadas por hora de trámite.
- E. **Limpiar filtros:** vuelve a mostrar la lista de pedidos para el año en curso.

Todos los filtros se deberán realizar mediante *LinQ to Objects*, y se deberá usar *una* sola colección de objetos, sobre la cual se efectuarán los filtros.



## Funcionalidades Mínimas del Sitio Web

### Formulario Web: Master Page General

**Actor Participante:** Publica

**Resumen:** Esta MP desplegará un menú único de navegación, el cual tendrá los siguientes puntos:

- Inicio (referirse al Formulario Web “**Principal**”). Este es el formulario que se despliega por defecto en la MP General.
- Consulta tramites disponibles (referirse al Formulario Web “**Consulta Tramites**”)
- Registro (referirse a Formulario Web “**Registro de solicitante**”)
- Logueo (referirse al Formulario Web “**Logueo de solicitante**”)

### Formulario Web: Principal (formulario por defecto del sitio)

**Actor Participante:** público

**Resumen:** En este formulario se desplegara una breve descripción de sitio.

### Formulario Web: Consulta Tramites

**Actor Participante:** público

**Resumen:** En este formulario se desplegaran todos los tipos de trámites **que hay disponibles** (solo nombre). Si se selecciona un tipo de trámite, se deberán desplegar todos sus datos (incluyendo toda la documentación exigida para su realización). El despliegue de tipos de trámites deberá realizarse en un control **Repeater**. El despliegue individual deberá desplegarse en un control **XML**, formateado mediante **XSLT**. El formulario recibe un documento con formato XML (alojado en la memoria, **no físicamente**), el cual contendrá un nodo por cada tipo de trámite disponible, con todos sus datos. Dicho documento se deberá generar en la operación correspondiente en el Servicio del sistema. Además, se deberá poder filtrar los tipos de trámite por precio o por año (incluido en el código): el resultado del filtro se mostrara en el mismo control que la lista completa. El despliegue de los tipos de trámites y los filtros propuestos, deberán realizarse mediante **Link to XML**.

### Formulario Web: Registro Usuario Solicitante

**Actor Participante:** publico

**Resumen:** Este formulario permite realizar únicamente el alta de un usuario de tipo solicitante. Tomar en cuenta los datos necesarios que se especificaron en la sección “**Idea General**”. Además, se deberá crear en el Servidor SQL, un usuario de logueo y un usuario de base de datos, asociado a dicho usuario. Estos serán utilizados cada vez que el usuario quiera acceder a la base de datos. Tomar en cuenta la seguridad y acciones que debe realizar un usuario tipo solicitante, para la determinación de permisos.

### Formulario Web: Logueo

**Actor Participante:** público

**Resumen:** Mediante esta página se permitirá el logueo de un usuario solicitante al sitio (ingreso de cedula y contraseña). Si el empleado se autentica correctamente, el sistema lo re direccionará a la **página de solicitud de trámites**.

### Formulario Windows: Solicitud de Tramites

**Actor Participante:** solicitante

**Resumen:** Este formulario permite realizar el alta de una solicitud de trámite. De cada solicitud se deberá saber: numero interno (se genera automáticamente y es el identificador único), fecha y hora para realizar el trámite, y tipo de trámite a realizar. Por defecto, el estado de una solicitud es **Alta**. Además, cada solicitud deberá saber que usuario la genero.



---

## Funcionalidades del Servicio Windows

Se deberá generar un instalador de la aplicación de escritorio, el cual también instalara un servicio Windows en la estación de trabajo.

Dicho servicio deberá constantemente verificar el archivo XML de logueo, generado en el directorio de la aplicación. Si el archivo existe, deberá comprobar la hora actual con el horario de finalización de tareas del usuario logueado (ambos datos presentes en el archivo).

Si el usuario esta en horas extras, deberá generar un registro en la base de datos con la fecha y cantidad de minutos extras generado. Si este último registro ya existiese, se deberá actualizar la cantidad de minutos extras.





## Requerimientos de Implementación

- Implementación completa del sistema con tecnologías .NET en lenguaje C#. Obligatorio la entrega en **Visual Studio 2015**.
- La información deberá almacenarse obligatoriamente en una base de datos **SQL Server 2008 R2**
- El script de la base de datos debe generarse *manualmente*, sin la ayuda de un asistente. Deberá contener el *Esquema de creación de la base de datos*, *Stored Procedures* necesarios para realizar todas las tareas solicitadas, *Creación de usuarios y permisos* solicitados y necesarios, e *Inserción* de datos de prueba. Las restricciones sobre datos, deberán realizarse en la propia estructura de la tabla (uso de los modificadores *unique*, *check*, *default*).
- Las eliminaciones a nivel de la base de datos deberán ser físicas o lógicas en función de las posibilidades. Es decir, si un elemento a eliminar no tiene elementos dependientes dentro de los registros de la base de datos, o estos se pueden eliminar, se elimina físicamente; de lo contrario se hará una eliminación lógica. **Obligatorio de implementar**
- Para el desarrollo del sistema utilizar la arquitectura en 3 capas vista en el curso, mediante la utilización de bibliotecas de clases.
- **Obligatorio** el uso de clases definidas por el usuario para la comunicación de datos entre componentes (tanto para invocación como respuesta).
- Los componentes de Lógica y Persistencia deberán ser generados en base a los patrones de *Fábrica y Singleton* vistos en clase. No deberá implementarse ningún otro patrón.
- Siempre se deberá de trabajar en forma conectada (ADO.NET) para el manejo de la información de la base de datos.
- Los componentes deberán lanzar excepciones en caso de error. No se contempla otra forma de comunicación de errores entre componentes
- La comunicación del Servicio Web que publica las operaciones de la capa Lógica, deberá realizarse mediante contratos de servicio, **Windows Communication Foundation** (WCF).
- Deberán usarse **obligatoriamente Temas y/o CSS** para generar una interfaz gráfica uniforme en toda la web.

## Requerimientos de Entrega

- Solución completa del Software
- Script de la base de datos
- Modelo Conceptual.
- MER
- Diagrama de Clases completo de la Arquitectura en capas (incluye a todos los componentes, clases, interfaces y relaciones entre ellos)
- Diagrama de la Arquitectura del sistema
- Diagrama de Comunicación entre Componentes.

*Nota: todos los diagramas deberán ser generados con una herramienta para lenguaje UML. Estos deberán ser entregados en forma digital: una copia del archivo original del diagrama y una copia en formato PDF o JPG*



## Tutorías y Entregas

A continuación se detallan las entregas sugeridas:

- Primera Sesión (*semana del 20/01 al 24/01*):
  - Modelo Conceptual
  - Diagrama de Arquitectura
  - Diagrama de Comunicación de Paquetes
  - MER
  - Script BD
  - Interfaz gráfica web
- Segunda Sesión (*semana del 03/02 al 07/02*):
  - Sitio web totalmente funcional (sin Servicio)
  - Interfaz gráfica aplicación escritorio
- Tercera Sesión (*semana del 17/02 al 21/02*):
  - Servicio terminado
- Cuarta Sesión (*semana del 09/03 al 13/03*):
  - Entrega de todo lo solicitado en el punto **Requerimientos de Entrega**.
  - **Defensa obligatoria del proyecto**. Para ello el grupo deberá concurrir antes de su horario de tutoría, para realizar la instalación del sistema. Dicha instalación deberá respetar la arquitectura propuesta.