

# **Documento de arquitectura de Software**

**para**

**SR(Sistema de  
reclutamiento)**

**Versión 1.0  
aprobada**

**Preparado por  
Alexander Castillo Bruno  
Esteban Navarro Monge  
Jason Solano Fonseca**

**Laboratorio de Economía  
Experimental**

**19/04/17**



<b>Información General</b>	<b>2</b>
Alcance	2
Propósito	3
Usuarios deseados	4
Conformidad con esta práctica recomendada	4
<b>Referencias</b>	<b>4</b>
<b>Definiciones</b>	<b>4</b>
<b>Marco conceptual</b>	<b>6</b>
Descripción arquitectónica en contexto	6
Stakeholders y sus funciones	7
Actividades arquitectónicas en el ciclo de vida	9
Escenario: arquitectura de sistemas individuales	10
Escenario: arquitectura iterativa para sistemas evolutivos	10
Escenario: arquitectura de los sistemas existentes	10
Escenario: evaluación arquitectónica	10
Usos de las descripciones arquitectónicas	10
<b>Prácticas de descripción arquitectónica</b>	<b>11</b>
Documentación arquitectónica	11
Identificación de las partes interesadas y las preocupaciones	11
Selección de miradores arquitectónicos	11
Vistas arquitectónicas	11
Consistencia entre las vistas arquitectónicas	11
Razonamiento arquitectónico	11



# 1. Información General

## 1.1. Alcance

El alcance total del sistema SR 1.0 se puede apreciar en la siguiente lista de los módulos:

- Módulo usuario:
  - Módulo registro usuario:
    - Registro de los datos requeridos de un usuario.
    - Envío de correo de confirmación al registrarse un usuario.
  - Módulo inscripción experimento:
    - Envío de códigos correspondientes a la inscripción de un experimento.
    - Realización de la inscripción a un determinado experimento cerrado y abierto.
    - Visualización del horario de los experimentos que se realizarán.
    - Cancelación del experimento
- Módulo administrador:
  - Módulo control de usuarios :
    - Registro de un determinado usuario.
    - Eliminación de un determinado usuario.
    - Modificación de un determinado usuario
    - Habilitación y deshabilitación de un determinado usuario.
  - Módulo control de experimentos:
    - Registro de un determinado experimento cerrado, en el cual se genera un código para la inscripción.
    - Registro de un determinado experimento abierto.
    - Eliminación de un determinado experimento.
    - Modificación de un determinado experimento.
    - Filtración de los experimentos a los que se puede inscribir un



determinado usuario.

- Módulo control de correos:
  - Creación de la descripción de un correo.
  - Modificación de la descripción de un correo.
- Módulo consulta:
  - Filtración de datos acorde a la especificación del administrador.
  - Exportación de los datos a un archivo de Excel.

## 1.2. Propósito

El presente documento se realiza sobre el producto que recibe el nombre de Sistema de Reclutamiento (SR 1.0), el cual consiste en el mejoramiento y personalización de un sistema de reclutamiento llamado Orsee, el cual permita el ingreso, registro y modificación de usuarios (estudiantes universitarios, trabajadores, público en general), así también como diferentes experimentos que se vayan a realizar, llevando un mejor orden y administración de todas las actividades que se realicen.

Debido al ambiente de trabajo en el cual se desenvuelve, es de suma importancia para los administradores que el sistema pueda ser accesado desde cualquier lugar, para poder llevar un control exhaustivo de la información.

SR 1.0 permite el ingreso o registro de nuevos usuarios, como también experimentos y administradores, los cuales serán administrados por medio de un administrador Master.

SR 1.0 también permitirá el uso de credenciales para los administradores del sistema y definir permisos para el acceso de las funcionalidades del sistema. Para la parte de usuarios se permitirá el uso de credenciales por medio del correo electrónico registrado por cada usuario.

SR 1.0 debe procesar la información que posee de todas las actividades realizadas en el laboratorio de economía experimental y crear un documento csv con toda la información necesaria para llevar a cabo estadísticas por parte del administrador.





### 1.3. Usuarios deseados

Los principales usuarios que pueden hacer uso de este documento incluye a todos los interesados en la parte del desarrollo y la evolución del sistema, en este grupo se pueden incluir:

- Todas aquellas personas que posean o adquieran el sistema, como lo son los administradores del laboratorio, como también los asistentes del mismo.
- Todas aquellas personas que están inmersas en la parte del desarrollo, mantenimiento y entrega del sistema, como lo es el equipo de desarrollo.
- Todas aquellas personas que supervisan y evalúan todo el desarrollo del sistema, como lo pueden ser los asesores o auditores ya sean independientes o de la misma institución.

También se puede tener una clase secundaria de usuarios que pueden llegar a hacer uso del presente documento, este grupo comprende cualquier tercero que pueda estar involucrado en el mantenimiento, uso, o desarrollo del sistema en algún momento determinado.

### 1.4. Conformidad con esta práctica recomendada

Dentro de la arquitectura del software se presenta el patrón de diseño (Model View Controller) este patrón es utilizado para tener un gran control de datos y seguridad de los mismos. Ya que en la vista solo se presentan datos pero no se presentan las operaciones o funciones necesarias para obtener estos datos lo cual permite tener una gran prevenir métodos de sql injection, o similares para entrar o dañar datos en la base de datos. En el diseño de todas las funcionalidades existen tres elementos claves que son fundamentales para cumplir el patrón M.V.C estos elementos son:

- **Archivo con terminación .aspx:** en dicho archivo se especifica los diferentes elementos de la interfaz gráfica permitiendo al software obtener o presentar datos al usuario.



- **Archivo con terminación .aspx.cs:** en este archivo se presenta las diferentes funciones o procedimientos para controlar los datos que se utilizan en el software, también para administrar las diferentes conexiones que se realizan a la base de datos.
- **Archivo con terminación .aspx.designer.cs:** en este archivo se especifican todos los atributos utilizados en el archivo .aspx.cs permitiendo tener una gran representación de la estructura de la clase.

## 2. Referencias

El presente documento tiene como referencias la siguiente lista de documentos realizados con anterioridad para el desarrollo del sistema SR 1.0:

- Project Charter SR 1.0
- Visión y Alcance SR1.0
- Especificación de Requerimientos SR 1.0

## 3. Definiciones

- **SR1.0:** Siglas utilizadas para nombrar el sistema que se está desarrollando, las cuales significan “Sistema de Reclutamiento, versión 1”
- **Módulo:** Se puede utilizar para nombrar una parte o sección independiente de un programa.
- **Base de datos:** Como su nombre se refiere es un lugar donde se almacena toda la información necesaria para el buen funcionamiento del programa.
- **Hardware:** Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.
- **Software:** Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.
- **Memoria RAM:** es la memoria principal de la computadora, donde residen programas y datos, sobre la que se pueden efectuar operaciones de lectura y escritura.



- **MB:** El megabyte (MB) es una cantidad de datos informáticos. Es un múltiplo del byte.
- **GB:** El gigabyte (GB) es una cantidad de datos informáticos. Es un múltiplo del byte.
- **Ghz:** es una abreviatura de Gigahertzio, que es una medida de frecuencia creada por el físico H.R. Hertz.
- **Dispositivo:** es un aparato o mecanismo que desarrolla determinadas acciones. Su nombre está vinculado a que dicho artefacto está dispuesto para cumplir con su objetivo.
- **IDE:** Un entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés Integrated Development Environment (IDE), es una aplicación informática que proporciona servicios integrales para facilitar al desarrollador o programador el desarrollo de software.
- **Testing:** El testing es una actividad desarrollada para evaluar la calidad del producto, y para mejorarlo al identificar defectos y problemas.
- **Testing de Software:** El testing de software consiste en la verificación dinámica del comportamiento de un programa sobre un conjunto finito de casos de prueba, apropiadamente seleccionados a partir del dominio de ejecución que usualmente es infinito, en relación con el comportamiento esperado
- **Terceros:** son personas extrañas al contrato o al proceso, pero que desarrollan gestiones en ellos por así haberlo solicitado las partes directas o por tener un interés en el acto o contrato.

## 4. Marco conceptual

### 4.1. Descripción arquitectónica en contexto

Con respecto al entorno de trabajo del sistema, el mismo estará ubicado en las instalaciones del laboratorio de economía experimental, el cual se encuentra ubicado en el edificio de la escuela de Administración de Empresas de la Sede Central del Instituto Tecnológico de Costa Rica, el mismo es completamente



independiente a cualquier otro sistema que se utilice en las instalaciones del laboratorio, dando la libertad de poder ser utilizado sin la necesidad de esperar respuestas o conexiones de otros sistemas para su utilización.

Con respecto a lo que es el entorno de hardware y software, el sistema necesitará de forma principal que las computadoras donde será ejecutado tenga acceso a internet y un navegador de web (el de su preferencia), para poder tener acceso al mismo, con respecto a las características de hardware que disponga de al menos 4096MB de memoria RAM, un procesador con una velocidad de al menos 2.2Ghz, y un almacenamiento principal de al menos 500 GB, con respecto al sistema operativo se deberá usar Windows, para su mejor ejecución la versión recomendada es Windows 10.

Existen varias preocupaciones que tienen los stakeholders con respecto al desarrollo, funcionamiento y mantenimiento del sistema, algunas de ellas son las siguientes: rendimiento, confiabilidad, seguridad, evolutividad, las cuales serán desarrolladas más adelante en el documento.

#### **4.1.1. Visión General:**

El sistema se puede dividir estructuralmente en cuatro grandes grupos o paquetes, los cuales a su vez poseen todas las clases y funciones necesarias para el adecuado funcionamiento del sistema.

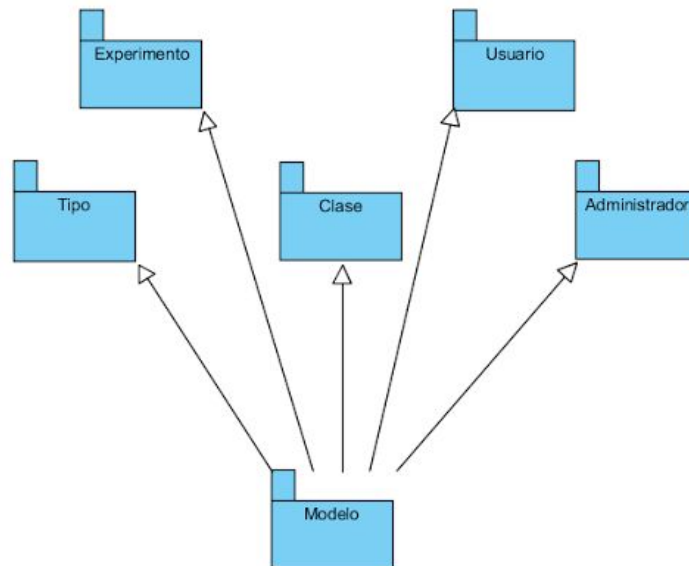
A continuación se describirán brevemente los diferentes grupos que lo conforma.





#### 4.1.1.1. Paquete de Modelo:

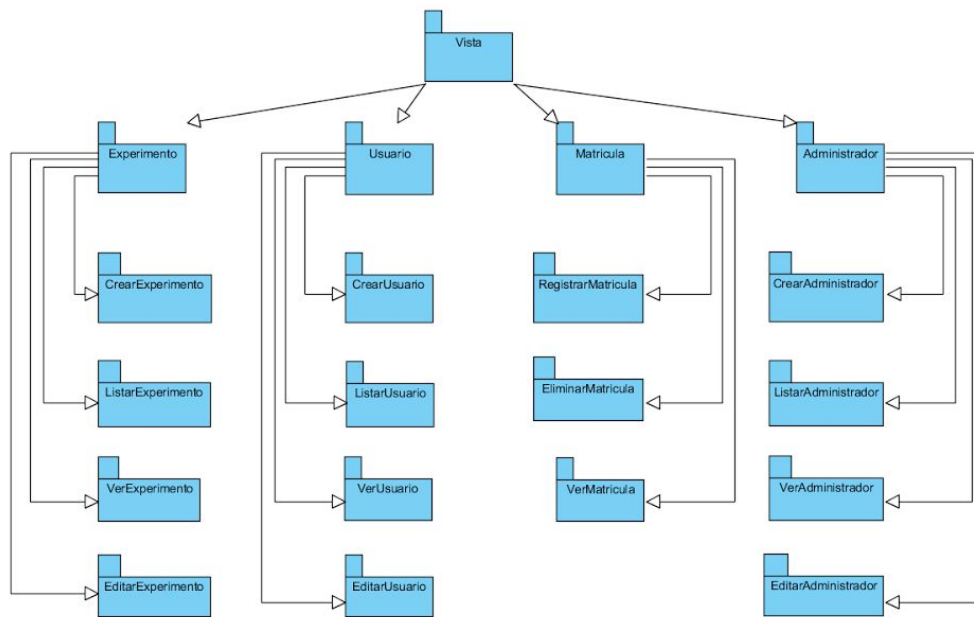
En este grupo se pueden encontrar todas las clases que son necesarias utilizadas para la representación de las diferentes entidades que se encuentran presentes en el sistema.



#### 4.1.1.2. Paquete de Vista:

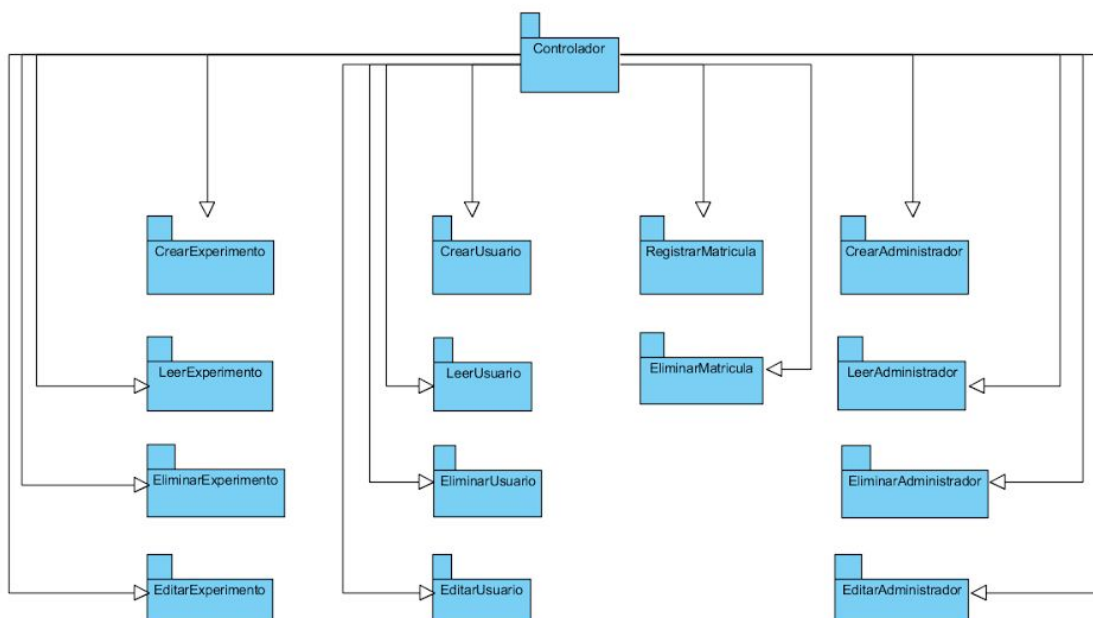
Como su nombre lo dice es el encargado de contener todas las clases que su funcionamiento consiste en renderizar todas las páginas a las cuales el usuario puede tener acceso.





#### 4.1.1.3. Paquete de Control:

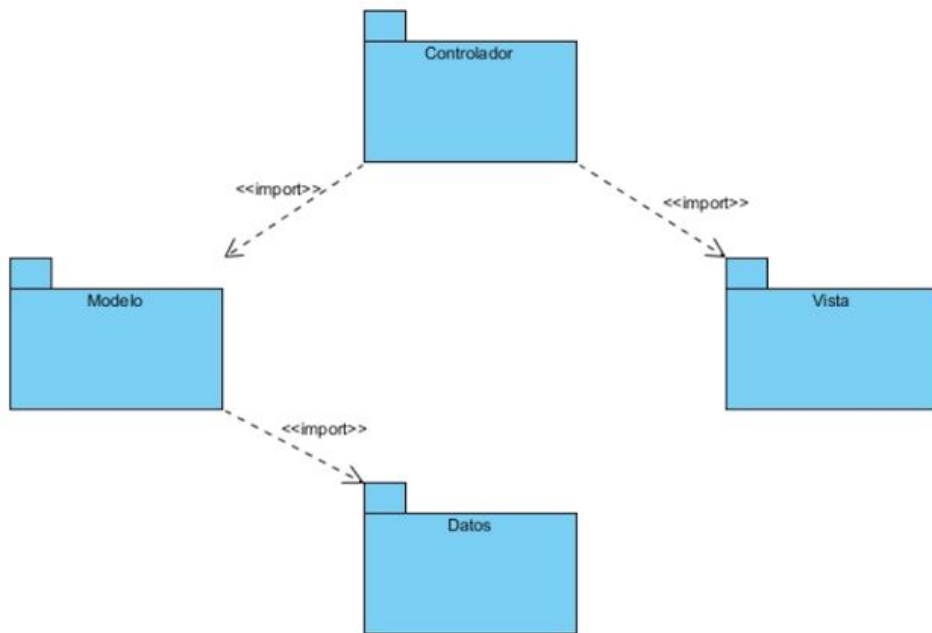
En este paquete se encuentran todas las clases encargadas de velar de que el funcionamiento de todos los requerimientos del sistema sea el correcto.





#### 4.1.1.4. Paquete de Datos:

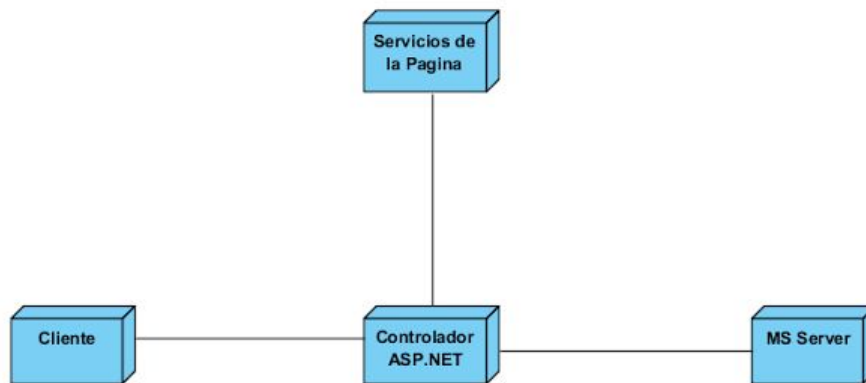
Dentro de este paquete se podrán encontrar todas las clases por medio de las cuales se tiene la comunicación con la base de datos y así poder tomar los datos necesarios para que sea desplegados en pantalla.





#### 4.1.2. Vista de despliegue:

Dentro de este paquete se podrán encontrar todas las clases por medio de las cuales se tiene la comunicación con la base de datos y así poder tomar los datos necesarios para que sea desplegados en pantalla.



**4.1.2.1. Clientes:** Se pueden conectar al sistema desde cualquier dispositivo que cuente con una conexión a internet y cuente con un navegador web. (Preferiblemente una PC o Laptop)

**4.1.2.2. Servicios de la Página:** Los servicios que incluyen son los siguientes:

- Registro, modificación y eliminación de usuarios.
- Registro, modificación y eliminación de Administradores, también la opción de recuperar contraseña.
- Registro, modificación y eliminación de experimentos.
- Matrícula a sesiones, tanto con invitación o sin invitación (sesiones abierta)
- Envío de correos de invitación.





**4.1.2.3. Controladores ASP .NET:** son los encargados tanto de la comunicación con la base de datos, como de realizar las diferentes modificaciones al HTML para desplegar toda la información que sea solicitada en pantalla.

**4.1.2.4. MS Server:** Es el motor de base de datos encargado de guardar todos los datos necesarios para que el sistema funcione de la forma correcta y muestra datos correctos.

### **4.1.3. Vista de desarrollo:**

Con respecto a las vistas de desarrollo, la misma abarca todo lo referente a los modelos de desarrollo, los modelos de implementación como lo son la programación por capas o subsistemas y las diferentes tecnologías implementadas en el sistema.

#### **4.1.3.1. Descripción general:**

El Sistema de Reclutamiento (SR.10) es implementado y desarrollado utilizando cuatro niveles de arquitectura, por medio del cual permite un desarrollo de tipo modularizado del sistema web, el cual nos ayuda a cumplir y seguir las diferentes normas de calidad de diseño que implica.

Los diferentes niveles de arquitectura ya fueron nombrados y explicados de forma breve en el punto 4.1.1 de Visión General, estos diferentes niveles también los podemos llamar como paquetes tal como se nombran en el punto mencionado.

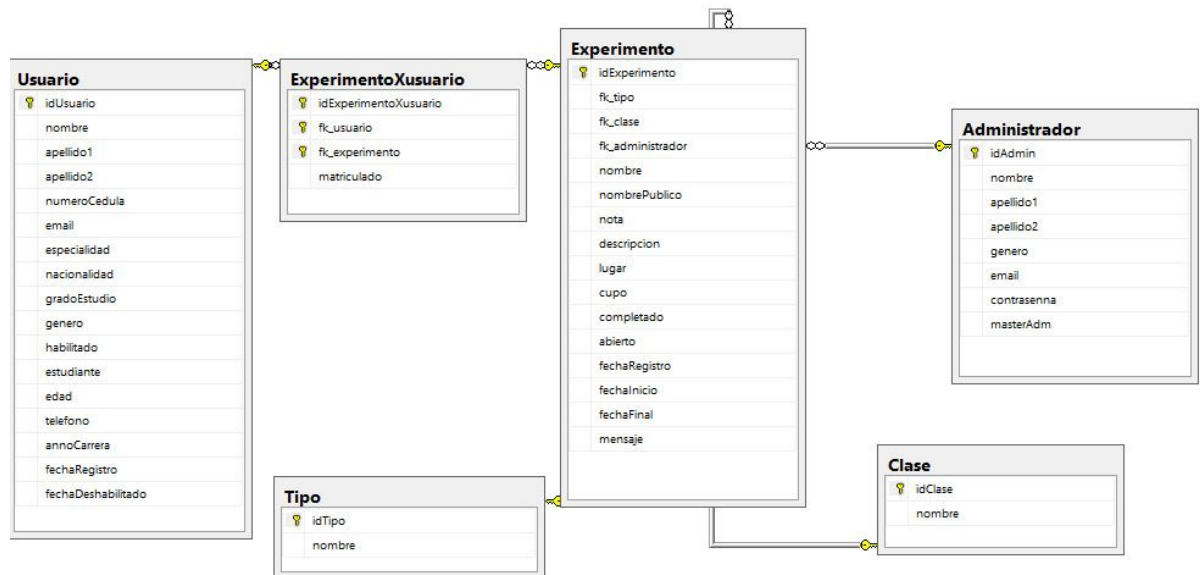
#### **4.1.3.2. Vista de Datos:**

En esta sección describe el sistema utilizado para la vista de datos el cual será el encargado de velar por la integridad y persistencia de los datos utilizados por SR 1.0.

El equipo de desarrollo decidió hacer la implementación de una base de datos relacional, la cual fue implementada en el motor de base de datos de Microsoft MS Server, la cual por medio de procedimientos almacenados, funciones y servicios web por medio de ASP.NET, sea posible mantener la persistencia de los datos que sean requeridos.



A continuación se presenta el diagrama de entidad relación de la base de datos:



## 4.2. Stakeholders y sus funciones

A continuación se mostraran los diferentes Stakeholders que tienen el Sistema de reclutamiento, y se explicará brevemente las diferentes funciones que el mismo tiene.

- **Cliente:**

- Las funciones más importantes del cliente, es proporcionar de manera clara y concisa los diferentes requerimientos que desea que el sistema tenga, además de esto debe mantener una excelente comunicación a lo largo del proyecto con el equipo de desarrollo, para de esta manera evacuar o solucionar alguna duda o problema que surja a lo largo del proyecto.
- Por otra parte debe tener el compromiso de estar presente en las diferentes reuniones que se establezcan para el informe de avances del proyecto y de esta manera dar la aprobación de los diferentes avances realizados.



- **Usuarios:**

- Con respecto a los usuarios una de las funciones más importantes a la hora del desarrollo del sistema, es que ellos son los encargados de realizar algunas pruebas, las cuales son de suma importancia para el equipo de desarrollo para tener una idea de que tan sencillo o amigable es el sistema con los usuarios.
- También es una forma de tener una crítica directa de los futuros usuarios finales del sistema, el cual es de gran ayuda para solucionar algunos problemas o tener una mejor idea de cómo piensan los usuarios respecto al sistema.

- **Desarrolladores:**

- Los desarrolladores tiene una serie de funciones de suma importancia, ya que en este caso no solo cumple el rol de desarrolladores, sino también deben realizar el rol de Administradores de proyecto, arquitectos, diseñadores, testers y de hasta auditores.
- En la parte de administradores de proyecto, deben tener la capacidad de manejar de la forma más ordenada y clara todo lo relacionado con el proyecto, como lo son todas las reuniones con el cliente o usuarios, realizar las diferentes minutas y bitácoras de todas las reuniones, como también encargarse de que todos los miembros realicen sus trabajos asignados en el tiempo debido.
- En el rol de arquitectos, se deben encargar de desarrollar y mantener una arquitectura para el sistema por medio de la cual se pueda satisfacer al cliente, para esto el arquitecto debe trabajar a partir de los requerimientos que fueron proporcionados por el cliente. También el arquitecto debe realizar la arquitectura como una especie de guía para que el grupo de desarrolladores comprenda de forma simple lo que deben realizar.
- En el rol de desarrolladores, pues como su nombre lo dice son los encargados de realizar que la arquitectura brindada por el arquitecto, puede tener toda la funcionalidad que desea el cliente, y así poder



cumplir con la expectativa del cliente. Son los encargados de toda la parte de programación interna del sistema.

- Con respecto a los diseñadores, trabajan en conjunto con los desarrolladores, ya que una de las funciones principales de ellos es asegurarse que el sistema tenga un ambiente amigable para el usuario y el mismo se puede desenvolver en el sistema sin ninguna preocupación.
- Por último en el rol de testers, la función más importante que tiene es el asegurarse de probar todas las funcionalidades realizadas por el grupo de desarrollo, y dar el visto bueno de que funcionan de la forma adecuada, asegurándose que el sistema trabaje de la mejor forma y no de ningún problema.

### **4.3. Actividades arquitectónicas en el ciclo de vida**

En cuanto a las actividades arquitectónicas realizadas por el equipo de desarrollo desde un principio se planteó utilizar el paradigma orientado a objetos con el fin de diseñar diferentes clases que puedan realizar las diferentes funciones que permiten solventar los requerimientos especificados por el cliente.

Por otra parte en cuanto a la toma de decisiones se implementa la metodología del desarrollo ágil por lo cual en cada ciclo iterativo se realiza una planificación, análisis de los requisitos correspondientes a los casos de uso que se resolverán en dicha iteración, diseño de estos casos de uso, implementación en código del diseño realizado y por último se realizan las pruebas y documentación de dichos casos de uso.

En cuanto a aspectos importantes del desarrollo del software es el cumplimiento de realizar diferentes módulos del sistema que sean reutilizables, esto quiere decir que del 40% o 60% de los módulos del sistema pueden ser reutilizables para implementar funciones similares en otros software permitiendo. A su vez esto permite mayor claridad al desarrollo del sistema y un código más entendible lo cual es muy favorable para el mantenimiento del software.





Además para las actividades detalladas de la ingeniería del sistema se tiene en cuenta los siguientes factores:

- **Objetivos y limitaciones de la misión:**

Antes de empezar el diseño para un caso de uso, se especifica con el cliente la meta de los requerimientos a realizar es decir el objetivo, y las restricciones de este caso se usó.

- **Concepto de operación:**

En esta actividad se describe las operaciones que realiza el sistema desde un punto funcional para un mayor entendimiento tanto para el cliente como para los desarrolladores.

- **Validar y verificar**

Esta actividad se realiza para asegurar que no existan inconsistencia entre los requisitos y el diseño arquitectónico. Por otra parte se planifican las pruebas físicas para que los cada requisito de un subsistema esté de acorde a los requerimientos.

- **Interfaces y ICD (Documento de Control de Interfaz)**

Para esta actividad se prepara un documento donde se especifican los diferentes tipos de interfaces, su funcionalidad e implementación en el código. Se debe de destacar que este documento se actualiza en cada revisión.

- **Gestión de la configuración y documentación**

Para la gestión de configuración y el control de la documentación después de terminar un baseline se deben de actualizar documentos importantes, por ejemplo manuales, diagramas de clases, diagramas de casos de uso, diccionario de datos y entre otros documentos donde se especifica la estructura del software, para esta implementación se utiliza las herramientas git y google drive.

#### **4.3.1. Escenario: arquitectura de sistemas individuales**

Toda la documentación presente el proyecto permite dar el conocimiento necesario a un programador que desarrolle en



.net. El sistema está desarrollado de tal forma que se pueden cambiar y agregar nuevas características en cualquier momento, todo esto de forma sencilla y sin poner en peligro la integridad de funciones o operaciones que realiza el sistema. Aparta los documentos donde se muestra la estructura del programa permite a los programadores tener mayor entendimiento al realizar mantenimiento o cambios al sistema.

#### **4.3.2. Escenario: arquitectura iterativa para sistemas evolutivos**

En cuanto a este escenario como se ha mencionado anteriormente al desarrollar el sistema se ha trabajado con un modelo iterativo en el cual se presentan varios casos de uso por iteración permitiendo al cliente solicitar cuales casos de uso pueden ser entregados por iteración.

Cada iteración es revisada por el equipo de desarrollo antes de ser entregada al cliente permitiendo tener un mayor análisis de los sistemas que son entregados, esto permite que las funcionalidades desarrolladas por cada caso de uso posean gran flexibilidad y adaptabilidad.

#### **4.3.3. Escenario: arquitectura de los sistemas existentes**

En caso de este escenario es muy poco probable que exista debido a la gran cantidad de documentación que existe para el software que se está desarrollando. Además se decidió utilizar .NET y C# como las herramientas para desarrollar, las cuales tienen constante soporte por Microsoft, e incluso documentación oficial de Microsoft en español brindando una forma concisa y segura de un gran respaldo para la vida del software y como se menciona anteriormente permitiendo a



desarrolladores futuros realizar mantenimiento y cambios al sistema desarrollado de una forma rápida y segura.

#### **4.3.4. Escenario: evaluación arquitectónica**

En secciones anteriores como 4.3.3 se han mencionada que el equipo de desarrollo realiza pruebas constantemente antes de entregar los casos de uso correspondientes a una iteración, es decir se realizan constantes pruebas para confirmar la calidad y integración de la funcionalidades desarrolladas. Por otra parte se tiene que tener la aprobación del cliente para que un caso de uso está completamente entregado esto quiere decir que el cliente evalúa los casos de uso con los parámetros que el sistema tendrá cuando se encuentre liberado. Todos estos pasos se realizan con el fin de tener un sistema evaluado constantemente.

### **4.4. Usos de las descripciones arquitectónicas**

Las descripciones arquitectónicas son de gran importancia en el desarrollo de un sistema, ya que en ella se describe el funcionamiento desde el punto de vista de arquitectónico, esto quiere decir desde un punto de vista estructural, como esta estructurado o formado, que tecnologías son utilizadas y porque, como está siendo implementado, estos son factores muy importantes a la hora de desarrollar el sistema, ya que al tener una buena arquitectura diseñada, la implementación de la misma podrá ser realizada sin muchos inconvenientes.

Las descripciones no solo nos ayudan a la hora de llevar a cabo la implementación del sistema, sino también puede ser utilizada para muchas actividades de tomas de decisiones como por ejemplo las que se listan a continuación:



- **Análisis de arquitecturas alternativas:** las descripciones arquitectónicas, ayudan para llevar a cabo análisis de que otros tipos de arquitecturas se pueden implementar en el proyecto ya sea para darle mantenimiento o mejorar su rendimiento.
- **Planificación de negocios para cambio de arquitectura:** tener una descripción de arquitectura sirve para poder llevar a cabo negociaciones para un cambio de arquitectura, si esta fuera necesario, y si una nueva arquitectura realizará las funcionalidades de una mejor forma.
- **Criterios de certificación:** Por medio de las descripciones arquitectónicas se pueden tomar cuales son los criterios de certificación de conformidad, con respecto a la implementación de la arquitectura.
- **Documentación de desarrollo y mantenimiento:** brinda una gran ayuda a la hora de desarrollar o dar mantenimiento al sistema, ya que por medio de las descripciones arquitectónicas se puede realizar el trabajo en el lugar específico que tiene que ser.
- **Revision, analisis y evaluacion:** contribuye de gran forma para inspeccionar y revisar que el sistema cumpla con el fin por el cual fue creado y de esta manera poder evaluarlo a lo largo de su vida útil.

Estos son los usos más importantes que se le pueden dar a las descripciones arquitectónicas, no obstante no son las únicas que existen, ya que se pueden utilizar para realizar mediciones o tener mejores visualizaciones para proyectos futuros.

## 5. Prácticas de descripción arquitectónica

### 5.1. Documentación arquitectónica

Fecha de emisión y estado:

La fecha actual es el 29 de abril del año 2017 y el estado del proyecto es casi listo, el proyecto se encuentra más o menos a un 95% de estar completo, falta 1 caso de uso por terminar y algunos problemas de UX por mejorar.





### Organización emisora:

No existe organización emisora del software, el mismo está siendo desarrollado como proyecto universitario para el curso de Proyecto de Software impartido por el profesor Dr. Jaime Solano

### Historial de modificaciones:

El historial de modificaciones ha sido vasto durante estos 2 meses de desarrollo del proyecto, entre los más importantes se encuentran. El cambio de los casos de uso en la aplicación móvil, los constantes cambios con respecto al tipo de datos almacenados en la base de datos, cambios en la organización de la información en la aplicación web principalmente la unión de las pestañas privacidad y reglas, entre otros.

### Resumen:

En resumen el proyecto consiste en desarrollar un sistema de reclutamiento que lo que pretende es la automatización del proceso de selección de posibles participantes de los experimentos organizados por el Laboratorio de Economía Experimental. Este software permitirá que cualquier persona que así lo desee pueda registrarse en el sistema y cuando un experimento sea creado se les notifique por medio de un correo electrónico. Seguido los posibles participantes podrán inscribirse en los experimentos que les interesan.

### Alcance del Proyecto:

Este fue definido en los anteriores documentos sugeridos al lector. El mismo no ha tenido ninguna modificación hasta el momento. Se le recomienda leer el capítulo 3 del documento de Visión y Alcance. Este detalla de forma específica el alcance y limitaciones de este proyecto.



## Contexto

El contexto del sistema se basa ciertos componentes los cuales son: La aplicación móvil, la aplicación web, la base de datos y los archivos de Excel los cuales genera el sistema. La mayor referencia de cómo se relacionan estos guiarse con el diagrama de contexto que se encuentra en el documento de Visión y Alcance.

## Glosario:

El glosario correspondiente este proyecto se encuentra en el documento ERS SR1.0.

## Referencias:

La lista de títulos de los documentos de referencia son los siguientes:

- Project Charter SR 1.0
- Visión y Alcance SR1.0
- ERS SR1.0

## 5.2. Identificación de las partes interesadas y las preocupaciones

Existen varios stakeholders los cuales se dividen en tres categorías:

**Usuarios del sistema:** Estos stakeholders van a ser todos los usuarios que se registren en el sistema, algunos ejemplos podrían ser estudiantes del Tec, profesores del Tec, estudiantes de colegio, empresarios, personas particulares, etc.

**Los adquirentes del sistema:** Estos son todos los administradores y personal del Laboratorio de Economía Experimental. Desde un punto de vista más general se podría decir que el principal adquirente del sistema es la Escuela de Administración de Empresas del Tec.



**Desarrolladores del sistema:** Los desarrolladores del sistema son Alexander Castillo Bruno, Esteban Navarro Monje y Jason Solano Fonseca

Existen varias preocupaciones que tienen los stakeholders con respecto al desarrollo, funcionamiento y mantenimiento del sistema, algunas de ellas son las siguientes: rendimiento, confiabilidad, seguridad, evolutividad.

El rendimiento es una preocupación de los stakeholders, ya que si el sistema no tiene un buen rendimiento puede generar que el sistema se vuelva lento y poco eficiente. Esta preocupación es un asunto complejo ya que los desarrolladores tratan de que el sistema sea lo más rápido posible, pero a la vez no se puede garantizar el buen rendimiento después de la fecha de entrega ya que una vez entregado el sistema los desarrollados no participaran en el proceso de mantenimiento.

La confiabilidad es un tema de suma importancia para cada uno de los stakeholders ya que por ejemplo los usuarios no desean de que su información sea revelada, debido a esto los administradores y desarrolladores buscan que la confiabilidad sea la más alta posible. La seguridad en este tema es prioritaria y por eso se desarrolla utilizando herramientas con medidas altas de seguridad.

El funcionamiento del sistema es una preocupación la cual se ha ido refinando durante todo el desarrollo del proyecto y los desarrolladores le hemos dado una prioridad alta, ya que se busca que todas las funciones del software tengan una actividad correcta. Esta funcionalidad se garantiza mientras tanto no se modifique ninguno de los factores ambientales como: disponibilidad del servidor, disponibilidad de la base de datos, entre otros.

La evolutividad dependerá del mantenimiento que se le dé al proyecto, como se mencionó anteriormente el mantenimiento no será responsabilidad de los desarrolladores sino que será responsabilidad del Laboratorio de Economía Experimental. Por esto aunque es una preocupación de los stakeholders pero no de los desarrollados.



### 5.3. Selección de los puntos de vista arquitectónicos

#### **Visión General:**

La visión general tiene como stakeholders a todas las personas involucradas en el desarrollo del software, como lo son usuarios, los adquirentes y los desarrolladores del sistema. Las consideraciones de esta visión son importantes ya que permiten toda la organización de la aplicación, en esta se muestran las diferentes partes de la aplicación, como lo son todas las tablas de la base de datos y cómo estas conforman el modelo, los controladores y las vistas. A esto se le llama el modelo MVC. Para guiarnos utilizamos textos como Swebok.

#### **Vista de despliegue:**

La vista de despliegue tiene como stakeholders a todas las personas involucradas en el desarrollo del software, como lo son usuarios, los adquirentes y los desarrolladores del sistema. Las consideraciones de esta visión son importantes ya que componen todos y cada uno de los componentes que integran el software que se está desarrollando y los presenta al usuario mediante información en el navegador. Los lenguajes utilizados para desarrollar este proyecto son C#, Javascript, CSS, HTML, ASP.Net y SQL. Las referencias utilizadas para el desarrollo de esta vista son: variadas, por ejemplo sitios web como: <http://stackoverflow.com/>, <https://msdn.microsoft.com/en-us/library/kx37x362.aspx> y <https://www.w3schools.com/>.

#### **Vista de desarrollo:**

La vista de desarrollo la misma abarca todo lo referente a los modelos de desarrollo, los modelos de implementación como lo son la programación por capas o subsistemas y las diferentes tecnologías implementadas en el sistema. El Sistema de Reclutamiento (SR.10) es implementado y desarrollado utilizando cuatro niveles de arquitectura, por medio del cual permite un desarrollo de tipo modularizado del sistema web, el cual nos ayuda a cumplir y seguir las diferentes





normas de calidad de diseño que implica. Los stakeholders más interesados por esta vista son los de desarrollo ya que ellos son los que utilizan esta vista. Para esta vista se utilizan softwares como MS SQL y Visual Studio. Las referencias utilizadas para el desarrollo de esta vista son: variadas, por ejemplo sitios web como: <http://stackoverflow.com/> y <https://msdn.microsoft.com/en-us/library/kx37x362.aspx>.

## 5.4. Vistas arquitectónicas

### **Modelo:**

Es la representación de la información con la cual el sistema funciona, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación. Envía a la vista aquella parte de la información que en cada momento se le solicita para que sea mostrada, esto principalmente es solicitado por un usuario del sistema. El modelo en esta aplicación está implícito en la base de datos. En nuestro sistema las solicitudes son realizadas por el usuario mediante la vista, luego la vista las transporta por medio de los controladores y estos determinan la información que se debe solicitar al modelo el modelo únicamente provee estos datos. Para esta vista arquitectónica utilizamos lenguajes como SQL (Stored Procedures y Queries) .

### **Vista:**

La vista corresponde a la información que se le presenta a los usuarios y administradores del software. Esta vista es desplegada en el navegador y puede ser accedida mediante cualquier dispositivo con un navegador integrado, pero se recomienda que en el caso de nuestra aplicación sea accedida por una PC o Laptop para una correcta visualización de todos elementos presentes en la aplicación. Esta representa la salida de los datos del Modelo y se los presenta de manera que el usuario encuentre la información que solicitaba. Para esta vista arquitectónica utilizamos lenguajes como HTML, Bootstrap y algunos objetos propios de ASP.

### **Controlador:**



Responde a eventos e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información. También puede enviar instrucciones a la vista asociada si se solicita un cambio en la forma en que se presenta el modelo por tanto se suele decir que el controlador es el intermediario entre la vista y el modelo. En nuestro sistema una vez que la vista recibe la solicitud por parte del usuario esta se encarga de pasarle la petición al controlador. El controlador recibe la petición y decide qué datos son los que se necesitan, pero como él no posee los datos envía la petición al modelo. Los controladores se pueden imaginar como las tuberías que transportan agua desde un centro de abastecimiento hasta la casa. Para esta vista arquitectónica utilizamos lenguajes como Javascript, Ajax y C#.

## 5.5. Consistencia entre las vistas arquitectónicas

La consistencia entre las vistas es imprescindible para que la aplicación funcione. Si una de las vistas antes mencionada no funciona correctamente puede generar errores como: datos incorrectos, datos extraviados, ejecuciones erróneas de la aplicación, fallos de seguridad, etc. La consistencia arquitectónica se trata de unir todos estos componentes y que los mismos interactúan entre sí de manera correcta. Es como unir piezas de un rompecabezas, para que todo el sistema funcione deben de estar las piezas acomodadas y diseñadas de tal manera que encajen con sus 'piezas vecinas'.

En la vista se puede notar que la consistencia es alta ya que existen mensajes claros que permiten alertar al usuario que el dato que ingreso es incorrecto y la razón. La información es presentada de forma ordenada y no se requieren múltiples pasos para cada acción del sistema. Los controladores realizan únicamente su trabajo de analizar y enviar información. Ya que si se les asignan otras funciones podría llevar a inconsistencias. Esta vista es de suma importancia y sin ella no aplicación no podría funcionar y generar múltiples inconsistencias. El modelo es donde se guarda la información, por lo que cualquier dato incorrecto en esta vista sería una inconsistencia segura en la aplicación. El modelo únicamente puede ser accedido por los controladores ya que si la vista accediera a él directamente no se podría determinar con certeza las acciones que se deben tomar para obtener la información necesaria.

## 5.6. Razonamiento arquitectónico



El patrón MVC fue utilizado ya que el proyecto así lo requería. Este patrón ayuda a mantener el orden y la modularidad. No se utilizó ningún otro patrón, ya que no se consideró necesario, tomando en cuenta que un patrón mal utilizado puede convertirse en un antipatrón. Algunos patrones creacionales que son muy comunes no aplican para este proyecto, como lo pueden ser el creacional o el prototipo.

Por otra parte no se consideró el uso del patrón adapter, ya que no se necesita utilizar adapter para conectar la base de datos, debido a que ASP.NET mediante un plugin de AWS realiza la conexión de manera muy simple. Otro patrón que se piensa que pudo haber sido utilizado es el mediator por la interconexión que da este patrón. Pero debido al análisis que hizo el grupo se concluye que no era estrictamente necesario. El último patrón que se consideró utilizar fue el patrón Proxy, este patrón puede haber sido utilizado en el ámbito de las conexiones a base de datos. Pero como ya se discutió con anterioridad el plugin de ASP.NET resuelve todo este tipo de problemas automáticamente.

