**Tecnológico de Monterrey**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Sistema de evaluación de solicitudes de beca**

Documento de Especificación del Diseño

**Dumplings**

A01176494, Fabiola Tamez

A01193575, Eduardo Hidalgo

A01193321, Jorge Amione

1.06

V 4.0

11/Nov/2020

**Control de Versiones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Versión | Descripción del cambio | Autor(es) | Revisor(es) |
| 08/09/202 | 1.0 | Creación del sistema | Jorge Amione  Fabiola Tamez  Eduardo Hidalgo | José Antonio Hernández |
| 1/10/2020 | 2.0 | Cambio de la entidad relación del sistema de becas, y del proceso. Se hicieron cambios en base a el nuevo Sprint 1 | Jorge Amione  Fabiola Tamez  Eduardo Hidalgo | José Antonio Hernández |
| 20/10/2020 | 3.0 | Se cambiaron los casos de uso ahora con la nueva información de parte de nuestros Stake Holders. | Jorge Amione  Fabiola Tamez  Eduardo Hidalgo | José Antonio Hernández |
| 11/15/2020 | 4.0 | Se agregaron nuevas referencias y guias de estilos. | Jorge Amione  Fabiola Tamez  Eduardo Hidalgo | Jose Antonio Hernadez |

**Contenido**

1. Introducción 4

1.1. Propósito 4

1.2. Alcance 4

1.3. Contexto 4

1.4. Resumen 4

2. Referencias 4

3. Glosario 4

4. Contenido detallado 4

4.1. Identificación de involucrados 4

4.2. Preocupaciones de diseño 4

4.3. Perspectivas 4

# Introducción

## Propósito

El propósito del proyecto es permitir que I2T2 tenga un sistema de solicitudes de becas para facilitar el manejo de las evaluaciones de los aspirantes por los usuarios del sistema (evaluadores). Por medio de este sistema se podrá dar calificaciones a los aspirantes y ayudará a obtener las estadísticas y datos necesarios para que el administrador decida si el aspirante merece la beca.

El sistema involucrará el manejo de los documentos que el CONACYT pide al I2T2 para las aplicaciones. De esta manera también se optimizará el proceso.

## Alcance

El sistema deberá permitir a los evaluadores acceder a una interfaz con un listado de solicitudes por área y componentes que permitan entregar una evaluación. De esta manera los datos necesarios de cada aplicación para estarán registrados para la toma de decisión final por parte del administrador. La interfaz tendrá sus órdenes, filtros y restricciones para un uso adecuado. El administrador deberá tener el poder de tomar la decisión final y de actualizar el contenido del sistema.

## Contexto

El sistema será una aplicación web en donde habrá dos tipos de usuario: administrador y evaluador. En este sistema se encontrarán todos los documentos de los aspirantes y al mismo tiempo el sistema permitirá a los evaluadores poder poner calificaciones a los aspirantes seguido por un resumen para el administrador de las calificaciones que se le dieron a los aspirantes.

El administrador deberá tener el poder de crear, editar y borrar aplicaciones, usuarios (evaluadores) y áreas.

## Resumen

El proyecto a realizar se trata de una aplicación web para el manejo de solicitudes de becas con dos tipos de usuario: los evaluadores, quienes interactúan con las aplicaciones de los aspirantes, y el administrador, quien otorga permisos, actualiza la plataforma (con nuevas aplicaciones y áreas) y maneja los datos generados.

# Referencias

* Presentación Power point por parte del cliente - <https://tecmx-my.sharepoint.com/:p:/g/personal/a01176494_itesm_mx/EYbd26_wyOxPtMHP8zhoYlYBM8SXA8kRpKlp59KI29Lf2A?e=xYLh0e>
* Se utilizara Heroku para poder hosting de la aplicación web la cual puedes encontrar informacion en <https://www.heroku.com>.
* Para el backend se utilizara PostgreSQL donde se puede encontrar informacion en https://www.postgresql.org.

# Glosario

I2T2 - Instituto de Innovación y Transferencia de Tecnología

Heroku – es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación.

# Contenido detallado

## Identificación de involucrados

* + José Antonio Hernández - Nuestro Cliente y Administrador de Evaluaciones
  + Jorge Amione – Administrador de Proyecto
  + Fabiola Tamez – SCRUM Master y Product Owner
  + Eduardo Hidalgo – Administrador de Configuración

## Preocupaciones de diseño

* + La principal preocupación de nuestros clientes es poder mantener un orden adecuado al momento de que los aspirantes son evaluados y al manejar los resultados.
  + Se necesita una interfaz con un listado de aplicaciones en un orden específico dado a las prioridades del cliente por área. Las aplicaciones deben estar ordenadas por menor número de evaluaciones, con un mínimo de 3 para ser procesada y con un máximo de 6.
  + Un evaluador solo debe ver las aplicaciones que involucren una de sus áreas asignadas.
  + El sistema debe proveer la calificación de los evaluadores por cada aplicación, así como sus comentarios, para la revisión del administrador.

## Perspectivas

### Perspectiva de contexto (PD001)

El usuario interactúa con el sistema para procesar las solicitudes de los aspirantes a una beca. El usuario puede ser administrador o evaluador.

### Vista de Administrador



El administrador entra al sistema y puede interactuar con él al aprobar perfiles de evaluadores y agregar solicitudes de aspirantes.

* El sistema le otorga un reporte general del procesamiento de becas por parte de los evaluadores, así como uno individual por cada aspirante.

El administrador puede hacer diferentes funciones con la aplicación:

* Crear solicitudes
* Aprobar evaluaciones

### Vista de Evaluador



El evaluador podra hacer dentro del sistema:

* Evaluar a los aspirantes ingresados por el administrador.

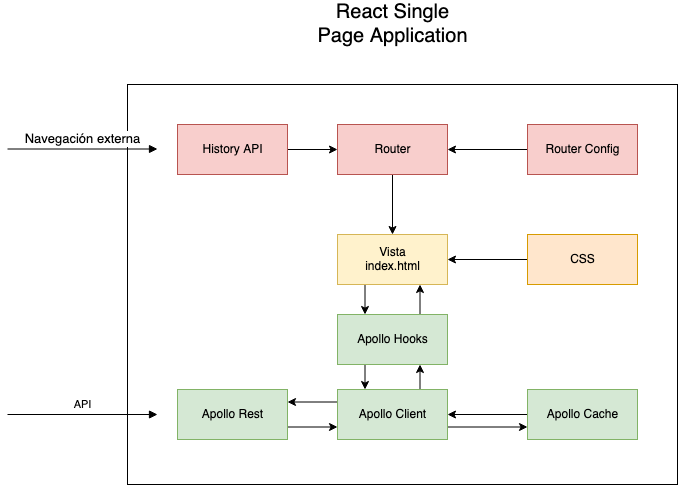
El sistema guarda la evaluación y despliega el resto de las aplicaciones disponibles.

### Perspectiva del desarrollador (PD002) (Arquitectura)

Las perspectivas del desarrollador se dividen en dos:

* + - La perspectiva del lado de desarrollo del cliente.
    - La perspectiva del lado de desarrollo de servidor.

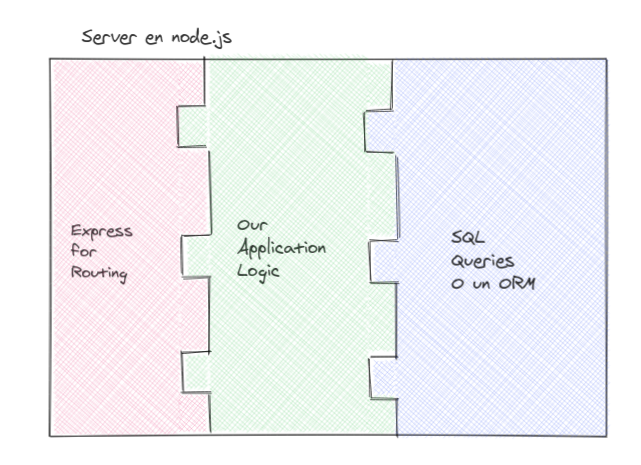
### Vista del cliente



La vista, la cual interactúa directamente con el usuario, es alimentada por varios componentes. Utilizando React, nuestra vista será un archivo HTML que se carga *client side*. Todo el JavaScript será cargado inicialmente, de acuerdo a lo que se necesite. Para navegación se utiliza el history API y es configurado por medio de React Router. Para manejar el estado y las llamadas a nuestro API, se utilizará la librería Apollo.

### Overview Arquitectura de nuestro servidor

Layers, Routing, Lógica de nuestra app (tan delgada como se pueda) pero que siempre este como intermediaria entre el routeo y la BD, y queries SQL o usar un ORM para accesar a postgresql

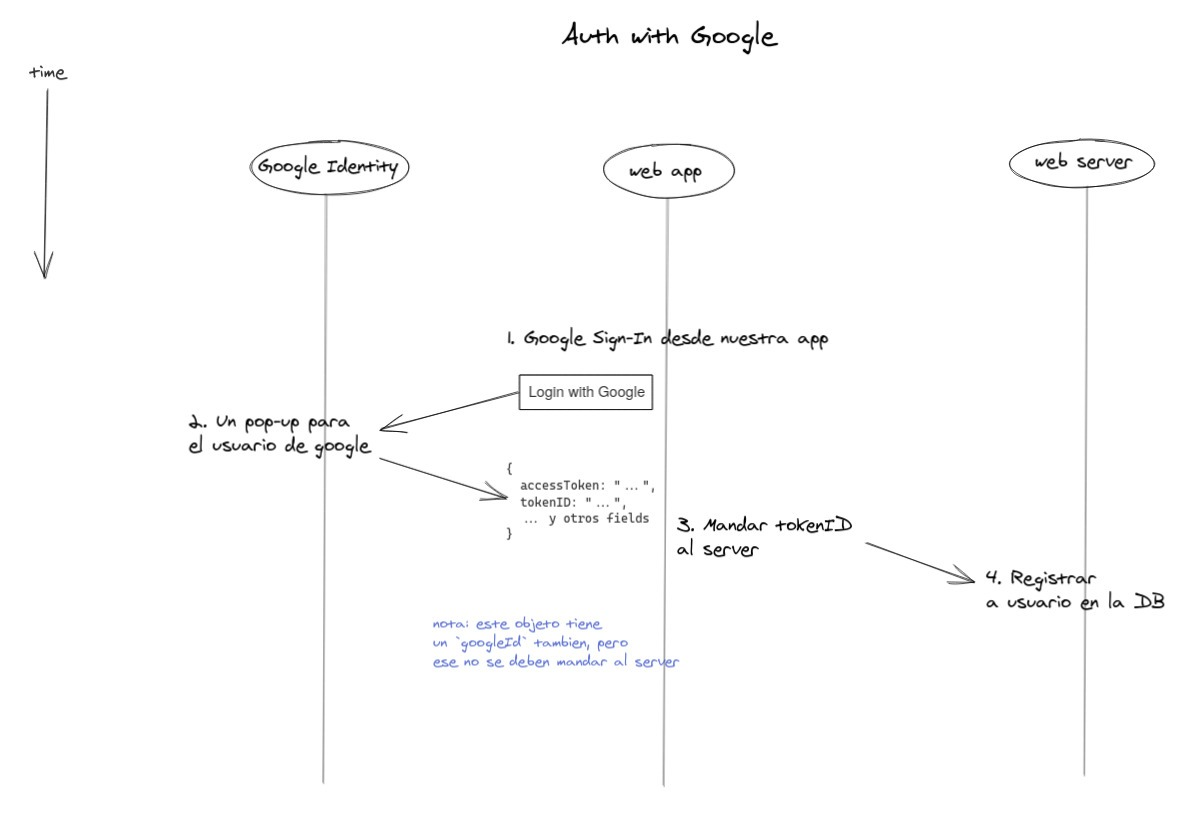


### Vista VD Diagrama entidad Relación del proceso

Entidades y sus relaciones del dominio del Sistema de becas, junto con notas/observaciones, así como unos pocos detalles de implementación.

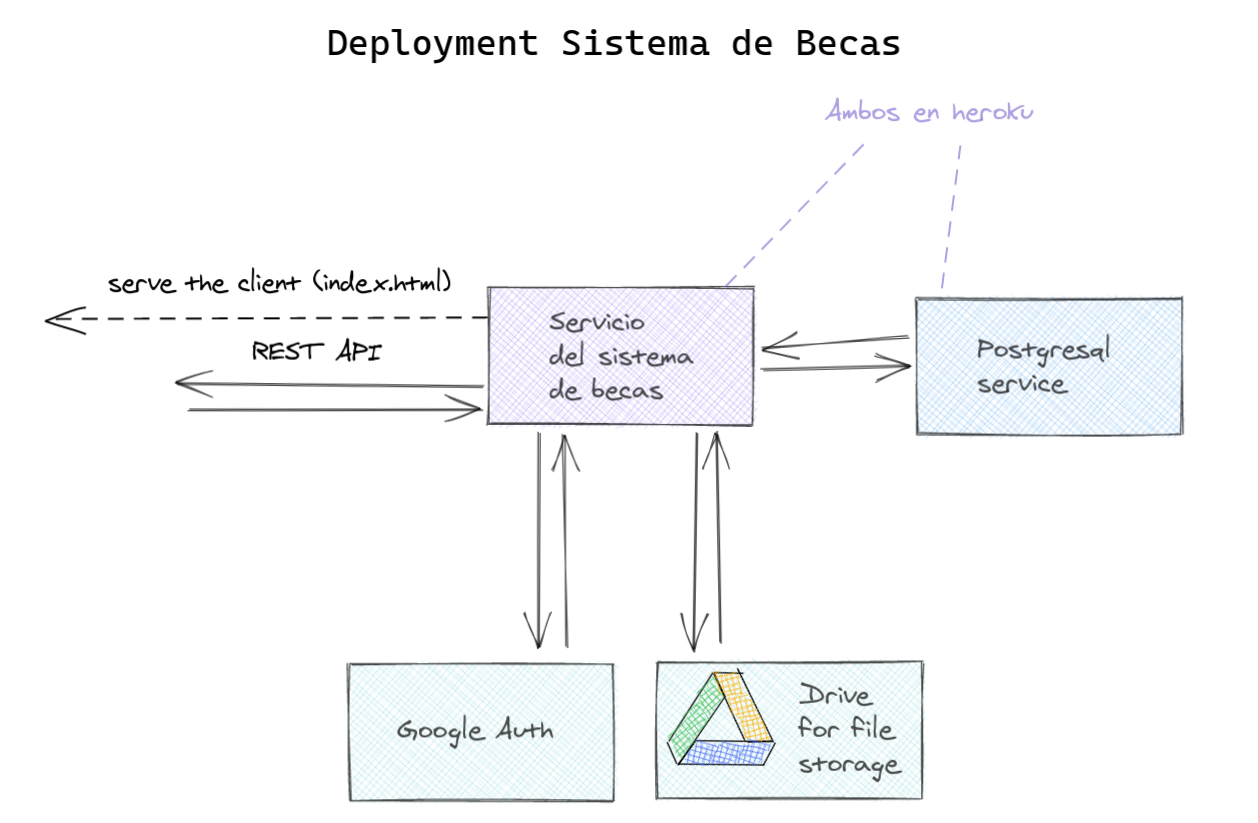
Diagram, schematic

Description automatically generated



### Perspectiva del despliegue (PD003) (Arquitectura)

### Diagrama de Despliegue

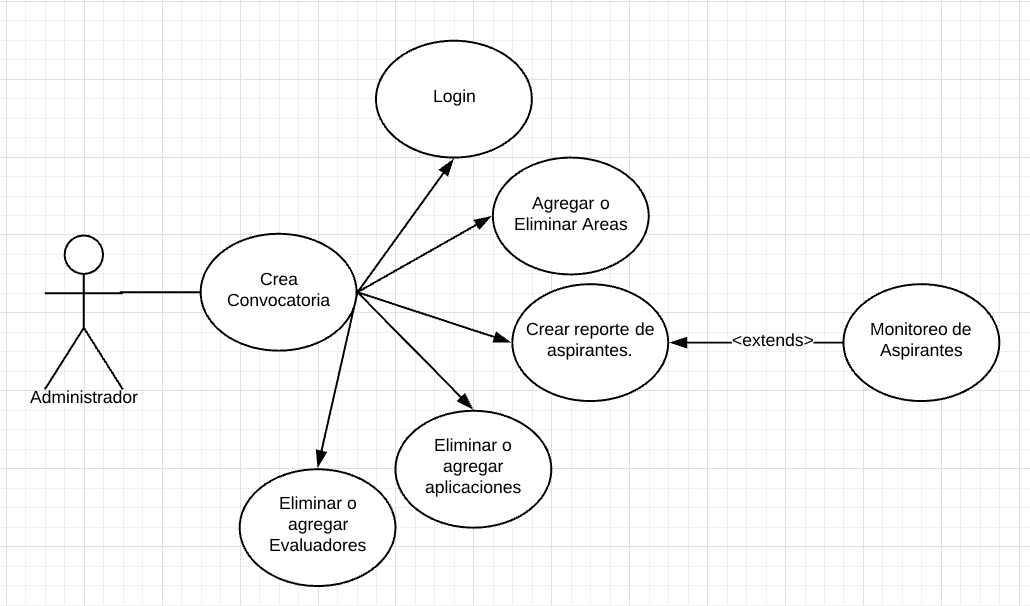


El sistema de manera general es un server en node.js, que corre en Heroku y que sirve el cliente y también expone un REST API para el funcionamiento de los clientes, una base de datos postgres que corre como servicio en Heroku también, e integraciones externas con Google para autenticación y drive para guardar archivos (y no tener que hacerlo nosotros

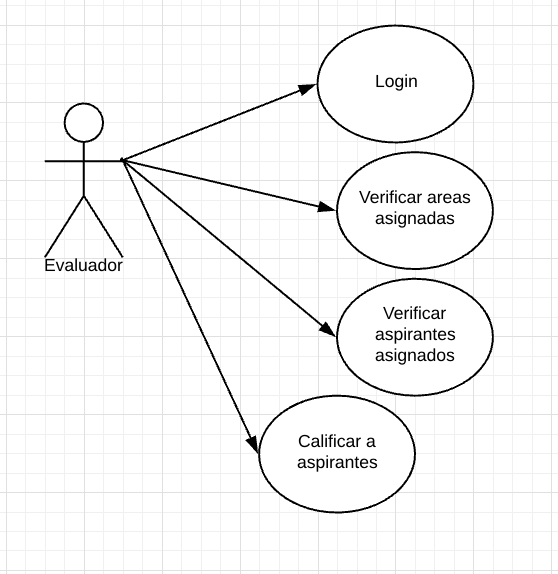
## Perspectiva del usuario (PD004) Casos de Uso

Hay dos tipos de usuarios para nuestro sistema uno siendo el administrador el cual se encarga de verificar las evaluaciones, crear y/o eliminar áreas y aplicaciones de becas. En el otro lado tenemos otro de nuestros usuarios que es el evaluador el cual puede tener acceso a sus áreas asignadas y de ahí poder ver los aspirantes a evaluar.

### Caso de Uso del Administrador



### Casos de Uso del Evaluador



### Guia de Estilos

El sistema contara con un dashboard diferente para el administrador y evaluador que sera fijo para amobos. Tendra un fondo blanco fijo con botones color azul y rojo con iconos de manera cuadrada se le agrega tambien sombra en la parte de atrás para dar a entender al usuario que es algo “clickable”.

Administrador contara con opciones como:

* Convocatorias
* Ver todas las solicitudes.

Evaluador contara con opciones como:

* Convocatorias Asignadas (Pueden ser multiples)

Diagram

Description automatically generated

A picture containing graphical user interface, text

Description automatically generated

Colores: Los colores de esta aplicación son basados en las recomendaciones de UX Design en base a los colores mas comodos para le usuario que son:

* Boton Primario: Rojo #d63d3b RGB (214,61,59)
* Iconos: Azul Base de Gradient #1e5799 RGB (30,87,153)
* Iconos: Azul Claro Base de Gradient #e4f5fc RGB (228,245,252)
* Botones: Gris claro #f3f0f0 rgb(243, 240, 240)

Font: Verdana

