



INFORME FINAL PRÁCTICA MODALIDAD EMPRENDIMIENTO

CULTURAL CITIES

JUAN CAMILO SOTO PINEDA
CARLOS ANDRES YEPES GALLEGO

DIRECTOR:
MAURICIO ARIAS JARAMILLO

TECNOLÓGICO DE ANTIOQUIA
INSTITUCIÓN UNIVERSITARIA
TECNOLOGÍA EN SISTEMAS POR CICLOS PROPEDEÚTICOS
MEDELLÍN, COLOMBIA.

2020

TABLA DE CONTENIDO

TABLA DE CONTENIDO.....	2
ÍNDICE DE FIGURAS	4
ÍNDICE DE TABLAS.....	6
ABREVIATURAS.....	7
1. INTRODUCCIÓN.....	8
2. PLANTEAMIENTO DEL PROBLEMA.....	9
3. OBJETIVOS	10
3.1. Objetivo General.....	10
3.2. Objetivos específicos.....	10
4. METODOLOGÍA.....	11
5. ESTRUCTURA DEL MODELO DE NEGOCIO	12
5.1. Segmentos del mercado.....	13
5.2. Estudio de competidores	13
5.3. Propuesta de valor.....	13
5.4. Fuentes de financiación.....	14
5.5. Alcance de la propuesta	14
6. ESTUDIO DE FACTIBILIDAD.....	15
6.1. Factibilidad técnica	15
6.2. Factibilidad económica	16
6.3. Factibilidad operacional	16
7. DESARROLLO TÉCNICO DE LA PRÁCTICA	18
7.1. Requisitos.....	18
7.2. Modelos de casos de uso	20
7.3. Diagrama de clases.....	26
7.4. Pruebas funcionales	27
7.5. Diseño de la interfaz de usuario.....	28
7.6. Despliegue del sistema.....	30
7.6.1. Manual de usuario	30
7.6.2. Manual de instalación	30
8. CONCLUSIONES.....	31

REFERENCIAS..... 32
ANEXOS 33

ÍNDICE DE FIGURAS

Figura 1. Diagrama de clases.....	26
Figura 2. Inicio App CulturalCities.....	28
Figura 3. Logueo App CulturalCities.....	28
Figura 4. Lista eventos App CulturalCities.....	29
Figura 5. Búsqueda eventos App CulturalCities.....	29
Figura 6. Registro usuario App CulturalCities.....	29
Figura 7. Despliegue del sistema.....	30
Figura 8. Inicio App Cultural Cities.....	33
Figura 9. Registro usuario App CulturalCities.....	34
Figura 10. Logueo App CulturalCities.....	34
Figura 11. Lista eventos App CulturalCities.....	35
Figura 12. Búsqueda eventos App CulturalCities.....	35
Figura 13. Panel de Control Azure.....	36
Figura 14. Panel de Control SQL Azure-1.....	36
Figura 15. Formulario creación SQL Database.....	37
Figura 16. Panel de Control SQL Azure-2.....	38
Figura 17. Panel de Control SQL Azure-2.....	38
Figura 18. Editor Consultas SQL Database.....	39
Figura 19. Editor Consultas SQL Database Subir Archivo.....	39
Figura 20. Editor Consultas SQL Database Subir Archivo-2.....	39
Figura 21. Editor Consultas SQL Database Subir Archivo-3.....	40
Figura 22. Tablas Base de datos.....	40
Figura 23. Acceso a Visual Studio 2019.....	41
Figura 24. Abrir proyecto.....	41
Figura 25. Seleccionar proyecto.....	42
Figura 26. Abrir Archivo de configuración.....	42

Figura 27. Modificar String de conexión.....	43
Figura 28. publicación en Azure-1.....	43
Figura 29. publicación en Azure-2.....	43
Figura 30. publicación en Azure-3.....	44
Figura 31. Configuración servicio Azure-1.....	44
Figura 32. Configuración servicio Azure-2.....	44
Figura 33. Configuración servicio Azure-3.....	45
Figura 34. Configuración servicio Azure-4.....	46

ÍNDICE DE TABLAS

Tabla 1. Metodología del proyecto.....	11
Tabla 2. RecursosTécnicos.....	15
Tabla 3. Factibilidad Económica – Inversión Diferida.....	16
Tabla 4. Factibilidad Económica – Inversión Fija.....	16
Tabla 5. Requisitos Funcionales y no Fionales.....	18
Tabla 6. Reglas de Negocio.....	19
Tabla 7. CU-01.....	20
Tabla 8. CU-02.....	21
Tabla 9. CU-03.....	22
Tabla 10. CU-04.....	23
Tabla 11. CU-05.....	24
Tabla 12. CU-06.....	26
Tabla 13. Pruebas Funcionales.....	27

ABREVIATURAS

APP: Aplicación

BD: Bases de Datos

C#: Lenguaje de programación de alto nivel

CSE: Custom Search Engine (Api de Google para la búsqueda personalizada)

RF-x: Requisito Funcional (x número consecutivo).

RNF-x: Requisito No Funcional (x numero consecutivo).

RN-x: Regla de Negocio (x numero consecutivo).

Cultural Bot: Bot encargado de realizar la búsqueda y análisis de nuevos eventos.

Cultural Daemon: Demonio encargado de ejecutar periódicamente Cultural Bot.

1. INTRODUCCIÓN

A lo largo de la vida académica, los alumnos adquieren conocimientos teóricos y técnicos sobre cada una de las áreas de su interés, sin embargo, carecen de la suficiente práctica o experiencia real dentro del mundo laboral, si bien, mientras el estudiante ejecuta sus estudios, se encuentra con diversos problemas prácticos planteados por sus profesores, tratando de llevarlos a entornos que permitan aplicar los conocimientos adquiridos en el análisis, planteamiento y desarrollo de la solución, no se logra simular suficientemente el entorno real del sistema productivo.

Es en este punto donde el modelo de práctica o emprendimiento, toma gran importancia, dentro de la formación de los estudiantes, ya que lo lleva a vivir de primera mano, todos los problemas que se pueden encontrar dentro de la vida productiva; sin embargo, para que este modelo cumpla exitosamente su objetivo, se debe replantear la forma de desarrollo, ya que el desarrollo de estas, debe generarse en el momento que el estudiante tenga las suficientes bases técnicas para cumplir a cabalidad con el objetivo.

Es necesario revisar el nivel de formación académica adquirido por el estudiante, para poder dimensionar la calidad del producto final esperado en la entrega, esto debido a que, en algunas de las áreas de estudio, tan solo en este último semestre se está abordando los temas técnicos necesarios para la implementación del proyecto final, y su desarrollo en paralelo, hace que se evidencie una de sus falencias, la cual es esperar a desarrollar el tema en el aula para poder aplicarlo en el proyecto.

Sin embargo, también es necesario destacar la necesidad de la metodología de práctica o proyecto de emprendimiento dentro de la vida académica de un estudiante, ya que, gracias a esta, el estudiante puede dar un primer vistazo de lo que lo espera en el mundo laboral, ya sea como empleado o como emprendedor, llevando a ubicar la materia como uno de los ítems más importantes en su carrera.

Con este proyecto se espera adquirir y afianzar todos esos conocimientos técnicos, que permitirán un buen desempeño del estudiante en el ámbito laboral y social, apoyando la estrategia con la investigación y trabajo en equipo.

2. PLANTEAMIENTO DEL PROBLEMA

Muchas veces se ha encontrado que han ocurrido eventos en la ciudad a los cuales la gente quisiera asistir pero se enteran de ellos solo después de finalizado el evento, al recorrer la agenda cultural de la ciudad, se encuentra que ocurren muchos eventos culturales sobre los cuales no se tienen pleno conocimiento, y su difusión por ser eventos no comerciales no es muy amplia, además, no existe una cartelera que reúna todos estos eventos en un solo lugar, por lo cual, se hace muy difícil que la gente este enterada de los eventos que ocurren en la ciudad de su preferencia.

Cultural Cities nace como una propuesta a la necesidad de las personas de consumir cultura en su entorno, permitiéndoles elegir sus gustos, almacenándolos en su cuenta personal y permitiendo así que Cultural Cities les ofrezca contenido web personalizado, siempre con las últimas novedades del mundo del entretenimiento para que los usuarios estén actualizados. El usuario recibirá alertas en su dispositivo móvil y también podrá ver estas novedades mediante un portal web transaccional donde podrá compartir sus gustos y opiniones sobre los eventos culturales como comentarios en el evento.

El alcance inicial de la aplicación está dado por la zona geográfica del área metropolitana del valle de Aburrá.

3. OBJETIVOS

3.1. Objetivo General

Proporcionar un sistema para los usuarios donde puedan consultar y compartir información cultural de sus alrededores.

3.2. Objetivos específicos

- Tener una aplicación móvil donde los usuarios puedan iniciar sesión y ver su configuración personal.
- Encontrar en una sola aplicación las noticias de mis categorías culturales preferidas.

4. METODOLOGÍA

Para el desarrollo del proyecto se implementó diferentes tecnologías de desarrollo buscando la optimización de cada uno de los procesos en la que intervienen.

Mediante el levantamiento de requisitos al inicio del semestre, se generó una ruta de desarrollo sólida, la cual permitió la ejecución del proyecto una vez fueron aprobados los requerimientos mediante la metodología de cascada, navegando en cada una de las fases definidas dentro del cronograma de trabajo.

N°	¿Qué?	¿Cómo?
1	Análisis del problema	<ul style="list-style-type: none">✓ Investigar productos similares en la red.✓ Analizar información obtenida y determinar el estado actual del mercado para el proyecto planteado✓ Generar propuesta diferenciadora frente a productos que puedan ofrecer algunas de las características planteadas para la solución al problema.
2	Estudio técnico	<ul style="list-style-type: none">✓ Identificar las plataformas que permitan realizar un sondeo de eventos en la red.✓ Identificar las mejores herramientas para satisfacer cada una de las necesidades que plantea la solución del problema.✓ Establecer la ruta de aprendizaje del grupo para capacitarse en las herramientas y plataformas seleccionadas.
3	Análisis Financiero	<ul style="list-style-type: none">✓ Identificar los costos fijos y variables de las tecnologías seleccionadas y capacitaciones necesarias para el grupo.✓ Establecer el método de financiación del proyecto.✓ Determinar el método a utilizar para generar retorno

Tabla 1. Metodología del proyecto.

Una vez surgió el problema, se empezó el análisis y ajustes por parte del equipo, teniendo como base el marco que el mismo problema generaba.

Se realizó investigación sobre las posibles fuentes o páginas que permiten suministrar soluciones cercanas al problema planteado, encontrando grandes oportunidades en el ámbito cultural.

Luego de analizar todas las variables que la solución a este planteaba, se genera una lluvia de ideas que permite encontrar la mejor ruta para la solución de este. Logrando de esta forma, trazar un plan de trabajo que permite la ejecución de la solución, genera un flujo ininterrumpido durante las siguientes semanas de desarrollo, abordando los requisitos de acuerdo con el plan de trabajo elaborado con anterioridad.

Lograr consolidar este flujo de trabajo, permitió la búsqueda de las mejores tecnologías para abordar cada uno de los módulos que dieron solución al problema.

Para dar solución a la búsqueda de los eventos culturales, se toma la decisión de utilizar el api de Google custom search, la cual permite realizar búsquedas en diferentes sitios, trayendo los resultados en un formato de fácil lectura para un computador, mediante la búsqueda de palabras claves y preferencias entregadas por el usuario final de la aplicación, el sistema permite por medio de este api de Google personalizar las búsquedas logando así, una mayor precisión en los resultados.

Sin embargo, la cantidad de información generada por las búsquedas de todas las preferencias posibles que permite seleccionar la aplicación, lleva al grupo a tomar la decisión de adoptar una de las tecnologías estrella en el tratamiento de grandes cantidades de información, y para esto se emplea Python, la cual permite recorrer y analizar dentro del mundo de información capturada desde Google custom search nuevamente las preferencias del usuario, generando así una mayor precisión en el resultado, adicionalmente deja una puerta para futuras funcionalidades.

Toda esta información deber ser llevada de una forma clara y sencilla al usuario final, y para esto se implementan servicios web, escritos en otra de las tecnologías de desarrollo de gran escalabilidad como lo es c#, lo cual permite que esta información pueda ser consumida en una primera versión desde dispositivos móviles Android y permitiendo dejar la puerta para otros dispositivos y entornos web.

Esta aplicación móvil, es generada en Xamarin, lo cual permite nuevamente aprovechar la potencia de desarrollo de una gran plataforma detrás.

Si bien se encuentran con limitantes de conocimiento a la hora de desarrollar el proyecto, el enfoque entregado al análisis y diseño del sistema apoyado en la correcta planificación permitió la adopción de las mejores tecnologías para su desarrollo, ganando una gran comunidad de apoyo y rica documentación sobre los diferentes problemas que se fueron presentando en el camino.

5. ESTRUCTURA DEL MODELO DE NEGOCIO

Cultural Cities nace como una propuesta a la necesidad de las personas de consumir cultura en su entorno, permitiéndoles elegir sus gustos, almacenándolos en su cuenta personal y permitiendo así que Cultural Cities les ofrezca contenido web personalizado, siempre con las últimas novedades del mundo del entretenimiento para que los usuarios estén actualizados

5.1. Segmentos del mercado

El sistema está planificado para satisfacer las necesidades de eventos culturales de las personas naturales. En el ámbito empresarial podrían beneficiarse los empresarios de eventos privados y públicos al poder promocionar sus eventos mediante el servicio de publicidad en la plataforma

5.2. Estudio de competidores

Google Shoelace (<https://shoelace.nyc>) es una app que tiene características similares que pretende solucionar las necesidades a las cuales apunta este proyecto, aunque su ámbito de aplicación está actualmente limitado a la ciudad de Nueva York.

A nivel nacional, se identificó a las compañías TuBoleta y BoletaEnMano que proporcionan información de eventos mediante correo electrónico cuando uno se ha inscrito en sus portales después de usar sus servicios de boletería, sin embargo, éstos no tienen el servicio de guardar las preferencias de eventos del usuario y se limitan a notificar todo lo que está en su cartelera activa

5.3. Propuesta de valor

Cultural Cities se diferencia de sus competidores fundamentado en los siguientes aspectos:

- De forma autónoma está sondeando páginas con información de eventos, registrándolos y proporcionando dicha información en tiempo real a los suscriptores de la aplicación.
- Proporcionará un nuevo canal a los empresarios de eventos para que puedan llegar a un público más amplio.
- Permitirá compartir opiniones sobre eventos mediante comentarios a éstos.
- Permitirá crear grupos entre usuarios con gustos afines.
- No está limitado a una plataforma de compras, no será un competidor de estas, sino, un aliado.

5.4. Fuentes de financiación

En la App y en el portal transaccional se incluirá servicios de publicidad de Google Ads con el objetivo de generar renta por la visita de los usuarios. Se buscará personalizar el contenido de las Ads relativo a eventos culturales, de manera que la publicidad no sea molesta para el usuario, sino que por el contrario amplíe el espectro de noticias que recibe el consumidor. En promedio un click en un ad puede generar entre \$1.000 y \$1.500 por lo que se debe garantizar un mínimo de 150 visitas y/o clics en los anuncios de la App para satisfacer el costo mínimo de usar la API “*Custom Search JSON API*” de Google en una fase inicial del proyecto (el costo de usar la API para más de 100 búsquedas diarias es de 5 USD por cada 1.000 *requests*, hasta 10.000 *requests* y un costo variable aproximado de \$175.000). Otras fuentes de ingresos adicionales son:

- Pagos de promotores de eventos culturales por publicidad específica de sus artistas representados.

5.5. Alcance de la propuesta

En su primera etapa, la aplicación se encargará de proveer la información correspondiente sobre eventos locales, tales como obras de teatro, danza, música, entre otros eventos culturales a los usuarios de la aplicación.

El sistema permitirá de esta forma, tener una lista de preferencias sobre las categorías de usuario, la cual permitirá mostrar solo la información relevante para el usuario, generando de esta forma una puerta de financiación por medio de la publicidad no invasiva focalizada, la cual dará garantía de que la publicidad mostrada en la aplicación será mínima y sobre eventos de su interés.

6. ESTUDIO DE FACTIBILIDAD

Dentro de la solución al problema se plantearon diferentes herramientas y estrategias para la puesta en marcha del proyecto, a continuación, podremos analizar la factibilidad técnica, económica y operativa de la propuesta de trabajo seleccionada.

6.1. Factibilidad técnica

La siguiente información corresponde a los requisitos técnicos para la puesta en marcha de la primera fase del proyecto en el área metropolitana.

Recurso	Cantidad	Especificaciones
Api Google: Custom Search Engine (CSE)	2	➤ menos de 100 consultas diarias.
Servidor de Base de Datos SQL Server	1	➤ 2 vCore. ➤ 10 GB RAM. ➤ 32 GB Almacenamiento.
App Service	1	➤ OS Linux. ➤ 4 vCore. ➤ 7 GB RAM. ➤ 10 GB Almacenamiento
Virtual Machines	1	➤ OS Linux. ➤ 4 vCore. ➤ 7 GB RAM. ➤ 285 GB Almacenamiento
PC portátil	2	➤ OS Windows. ➤ Intel Core I7. ➤ 16 GB RAM. ➤ 250 GB Almacenamiento SD
GitHub	2	➤ Team

Tabla 2. Recursos Técnicos

6.2. Factibilidad económica

Recurso	Cantidad	Valor Unitario (USD)	Valor Total (USD)
Api Google: Custom Search Engine (CSE)	2	0	0
Servidor de Base de Datos SQL Server	1	368.2	368.2
App Service	1	51.83	51.83
Virtual Machines	1	35.04	35.04
GitHub	2	4	8
TOTAL		459.07	462.07

Tabla 3. Factibilidad Económica – Inversión Diferida

Recurso	Cantidad	Valor Unitario (USD)	Valor Total (USD)
PC portátil	2	1398	2796
TOTAL		1398	2796

Tabla 4. Factibilidad Económica – Inversión Fija

6.3. Factibilidad operacional

La operación del proyecto es viable ya que, como valor agregado, al cliente personalizar sus preferencias, la aplicación solo le mostrara los eventos relacionados con esta, garantizando de esta forma una probabilidad de ingreso a la información del evento mayor que si se muestran eventos aleatorios, lo cual se traducirá en una mayor posibilidad de asistir al evento.

Partiendo de este principio de Cultural Cities, se buscará alianzas con los promotores de eventos culturales por publicidad específica de sus artistas representados, lo cual generara un mayor ingreso.

En la App y en su futura versión web se incluirá servicios de publicidad de Google Ads con el objetivo de generar renta por la visita de los usuarios. Se buscará personalizar el contenido de las Ads relativo a eventos culturales, de manera que la publicidad no sea molesta para el usuario, sino que por el contrario amplíe el espectro de noticias que recibe el consumidor, al no ser invasiva y ser tratada por la aplicación como nuevos eventos se garantizará que el usuario este cómodo con ella evitando retiros y ganando ingreso de nuevos usuarios gracias al voz a voz que se generará, el cual será nuestra principal estrategia de publicidad.

Además, Cultural Cities no espera ingresar como competencia para las principales tiendas de venta de boletas en línea, sino, que ingresara como un aliado estrategia para estas, al servir como búsqueda centralizada de los eventos que ofrecen cada una de ellas.

7. DESARROLLO TÉCNICO DE LA PRÁCTICA

7.1. Requisitos

ID	Nombre del Requisito	Descripción del Requisito	Prioridad	Usuario
RF-01	Búsqueda de Nuevos Eventos	Realizar Búsqueda de nuevos eventos por categorías y ubicación geográfica en la nube	Alta	Cultural Bot
RF-02	Búsqueda de Eventos	Realizar Búsqueda de eventos por categorías, palabras clave y ubicación geográfica en la aplicación	Alta	Usuario
RF-03	Ejecutar Cultural Bot	Ejecutar de forma periódica la tarea encargada de consultar nuevos eventos	Media	Cultural Daemon
RF-04	Registrar Usuario	Registro del usuario en la aplicación	Alta	Usuario
RF-05	Logueo de usuario	Logueo del usuario en la aplicación	Media	Usuario
RNF-01	Limite consultas Bot	Solo se pueden realizar máximo 100 consultas de nuevos eventos al día en CSE por parte de Cultural Bot	Media	Cultural Bot
RNF-02	Periodicidad Ejecución Cultural Bot	Se debe ejecutar (Cultural Bot) 4 veces al día la búsqueda de nuevos eventos	Media	Cultural Daemon

Tabla 5. Requisitos Funcionales y no Fionales.

Código	Nombre	Descripción
RN-01	validación nuevos Eventos	Al ejecutar la búsqueda de nuevos eventos el sistema debe tener como base la fecha de la última consulta, para evitar traer eventos repetidos, en caso de no encontrar eventos posteriores a la última consulta realizada no devolverá información.
RN-02	Control de duplicidad de los eventos	Antes de almacenar la información resultante en la base de datos, el sistema deberá validar que no existan eventos duplicados, de existir, no permitirá el almacenamiento del evento
RN-03	Periodicidad de ejecución Cultural Bot	Cultural Bot se debe lanzar cada 6 horas en búsqueda de nuevos eventos.
RN-04	Control de duplicidad del usuario	Al usuario realizar el registro, el sistema debe validar que este no se encuentre creado en la base de datos, de lo contrario, no debe permitir la creación del nuevo usuario. Esta validación se realizara con el correo electrónico
RN-05	validación de logueo	El correo electrónico y la contraseña deben coincidir con la almacenada en el base de datos y el usuario debe estar activo
RN-06	validación correo electrónico	El correo electrónico debe ser válido, para esto el sistema deberá enviar un correo de validación antes de continuar con el registro, para garantizar que la información almacenada corresponda al propietario del correo

Tabla 6. Reglas de Negocio.

7.2. Modelos de casos de uso

Tabla 7. CU-01.

Caso de Uso	
Realizar Búsqueda de nuevos eventos	
Código caso de uso	
CU-01	
Descripción	
Ejecución de Bot de búsqueda en la API Custom Search JSON API de Google, el cual se encargará de traer toda la información de los nuevos eventos.	
Actor	Descripción de sus responsabilidades
Cultural Bot	Ejecutar la búsqueda de nuevos eventos en la API Custom Search JSON API de Google
Flujo primario	
<ol style="list-style-type: none"> 1. Realiza la búsqueda de nuevos eventos. 2. Analiza la información encontrada en búsqueda de nuevos eventos. 3. Almacena la información en la base de datos. 	
Flujos alternos	
1. No hay eventos	
Reglas de negocio	
Los resultados de la búsqueda serán almacenados en la base de datos RN-01, RN-02	
Diagrama	

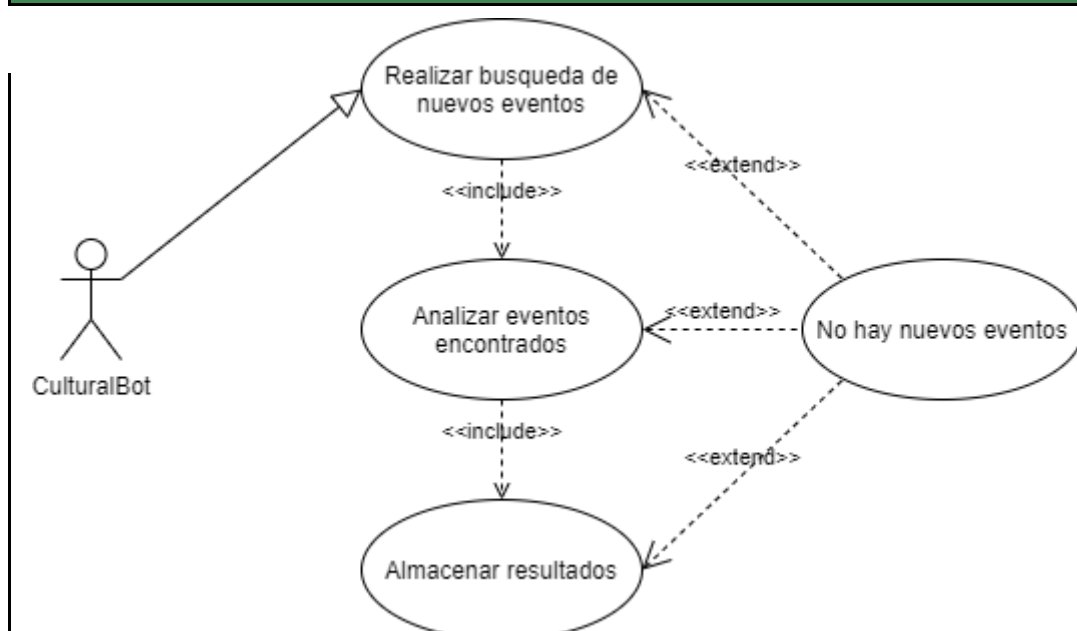


Tabla 8. CU-02

Caso de Uso	
Ejecutar Cultural Bot	
Código caso de uso	
CU-02	
Descripción	
Ejecución de Cultural Bot de forma periódica	
Actor	Descripción de sus responsabilidades
Cultural Daemon	Ejecutar la rutina que lanza la búsqueda (Cultural Bot)
Flujo primario	
1. Ejecuta la rutina de búsqueda.	
Flujos alternos	
No Aplica	
Reglas de negocio	
La rutina de búsqueda se ejecutará cada 6 horas RN-03	
Diagrama	

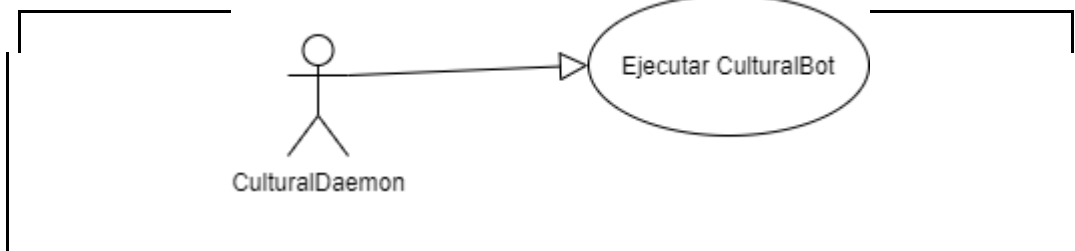


Tabla 9. CU-03

Caso de Uso	
Registrar Usuario	
Código caso de uso	
CU-03	
Descripción	
Registro del usuario en la aplicación.	
Actor	Descripción de sus responsabilidades
Usuario	Realizar registro con sus datos personales en la aplicación.
Flujo primario	
<ol style="list-style-type: none"> 1. Realizar registro. 2. Validar Usuario. 	
Flujos alternos	
<ol style="list-style-type: none"> 1. Validar Usuario bloqueado. 2. Validar Usuario no existe. 	
Reglas de negocio	
Para cada usuario se debe almacenar directamente la información en la base de datos RN-05, RN-06, RN-08	
Diagrama	

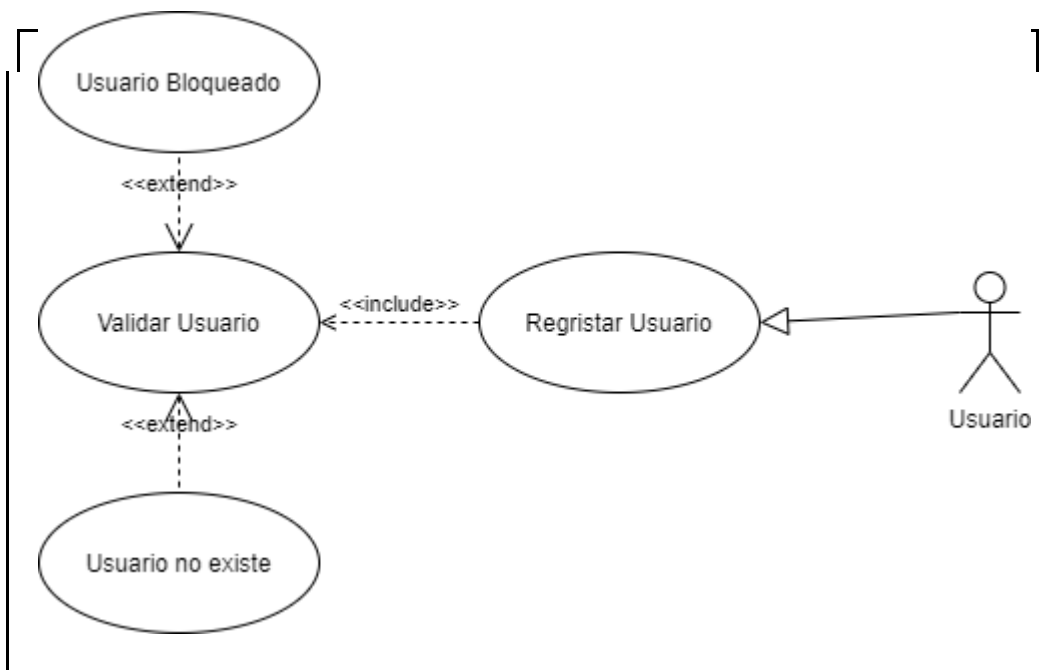


Tabla 10. CU-04

Caso de Uso	
Login	
Código caso de uso	
CU-04	
Descripción	
Logueo del usuario en la aplicación.	
Actor	Descripción de sus responsabilidades
Usuario	El usuario debe ingresar a la aplicación con su correo electrónico y contraseña.
Flujo primario	
<ol style="list-style-type: none"> 1. Loguear Usuario. 2. Validar Usuario. 	
Flujos alternos	
<ol style="list-style-type: none"> 1. Validar Usuario bloqueado. 2. Validar Usuario no existe. 	
Reglas de negocio	
Los datos deben coincidir con la información suministrada en la base de datos RN-07	
Diagrama	

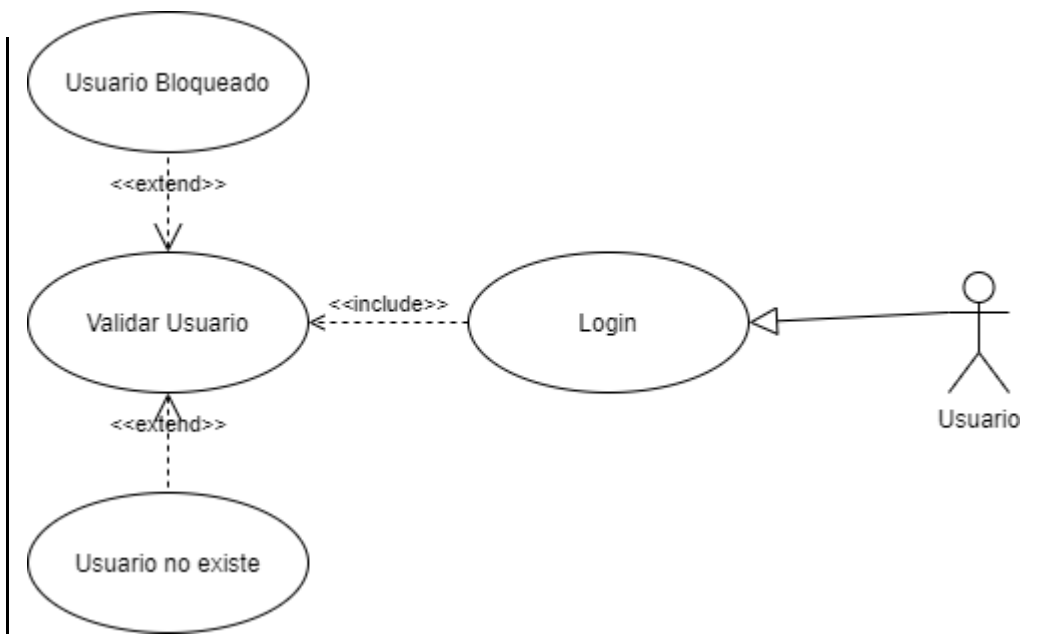


Tabla 11. CU-05.

Caso de Uso	
Modificar Datos Personales	
Código caso de uso	
CU-05	
Descripción	
Actualizar datos personales, registrar y modificar preferencias	
Actor	Descripción de sus responsabilidades
Usuario	Registrar y modificar sus preferencias y datos personales.
Flujo primario	
1. Modificar Datos Personales.	
Flujos alternos	
1. Usuario no logueado.	
Reglas de negocio	
No Aplica	
Diagrama	
<pre> graph TD Actor[Usuario] --> UC1(Modificar Datos Personales) UC2(Usuario No logueado) -.-> <<extend>> UC1 </pre>	

Tabla 12. CU-06.

Caso de Uso	
Buscar evento	
Código caso de uso	
CU-06	
Descripción	
Uso del motor de búsqueda de la aplicación para encontrar eventos por palabras clave o lugar geográfico	
Actor	Descripción de sus responsabilidades
Usuario	El usuario podrá realizar búsquedas por palabra clave o lugar geográfico.
Flujo primario	
1. Buscar evento.	
Flujos alternos	
1. No hay eventos.	
Reglas de negocio	
No Aplica	
Diagrama	
<pre> graph TD Usuario((Usuario)) --> BuscarEvento(Buscar evento) NoHayEventos(No hay eventos) -.-> <<extend>> BuscarEvento </pre> <p>The diagram shows a use case 'Buscar evento' with an actor 'Usuario' connected to it by a solid line with an open arrowhead. Below it, another use case 'No hay eventos' is connected to 'Buscar evento' by a dashed line with an open arrowhead and the label '<<extend>>'.</p>	

7.3. Diagrama de clases

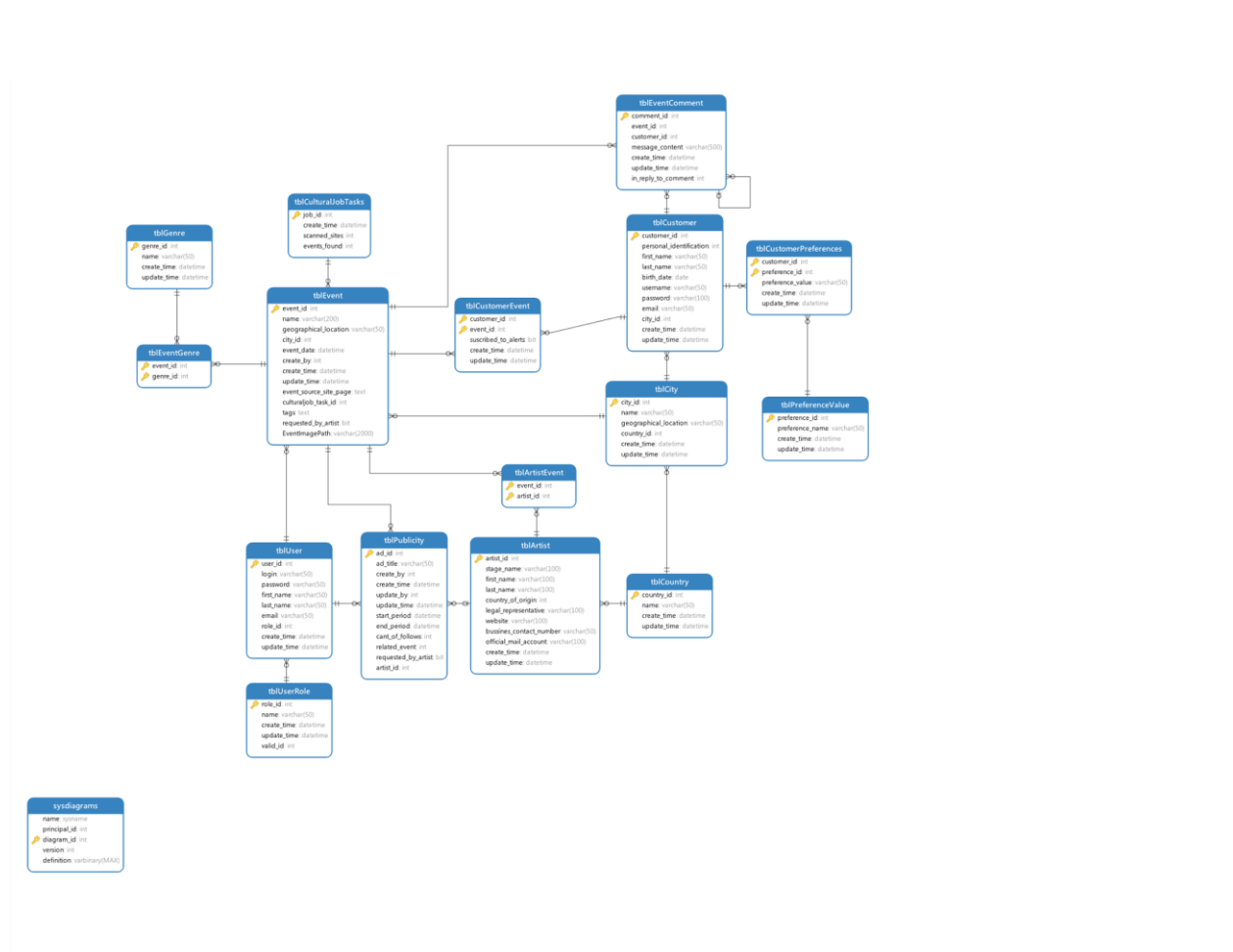


Figura 46. Diagrama de clases.

7.4. Pruebas funcionales

Tabla 13. Pruebas Funcionales

Fecha revisión	Módulo revisado	Observaciones	Resultados de la ejecución	Errores y éxitos detectados
2020-06-09	culturalSearch.py (Modulo de búsqueda en api de Google CSE)	Se realiza búsqueda de palabras clave (eventos+musica+medellin+2020) de en CSE	JSON con el resultado de la búsqueda	Errores = 0, la consulta trae los datos esperados
2020-06-09	culturalBot.py	Se realiza ejecución del Bot que se encargara de hacer las consultas en CSE	Error en inserción de eventos (no acepta parámetro tipo DateTime)	Errores = 1
2020-06-10	Modulo de login en App	Se realiza prueba de logueo con datos usuarios registrados en BD y no registrados	Login exitoso y login fallido para el caso del usuario no registrado	Errores = 0

7.5. Diseño de la interfaz de usuario



Figura 2. Inicio App CulturalCities.

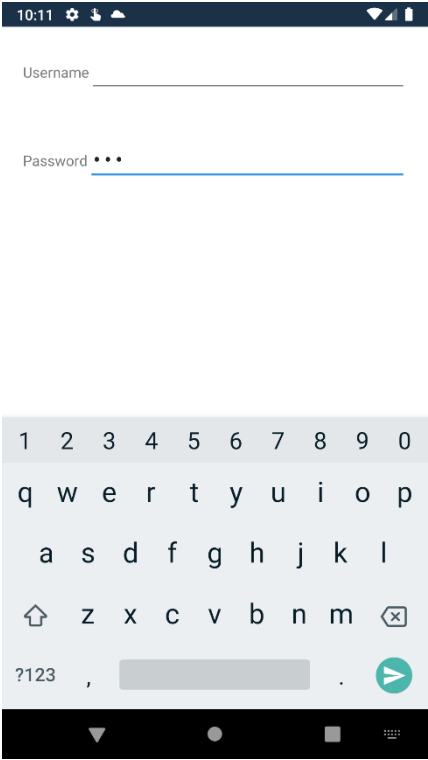


Figura 3. Logueo App CulturalCities

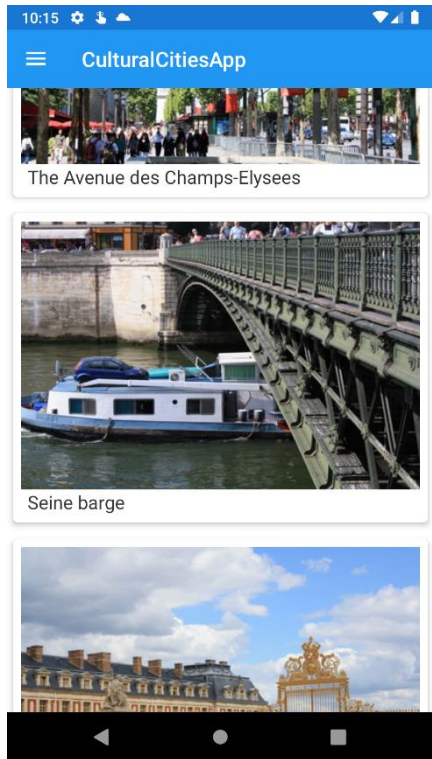


Figura 4. Lista eventos App CulturalCities.

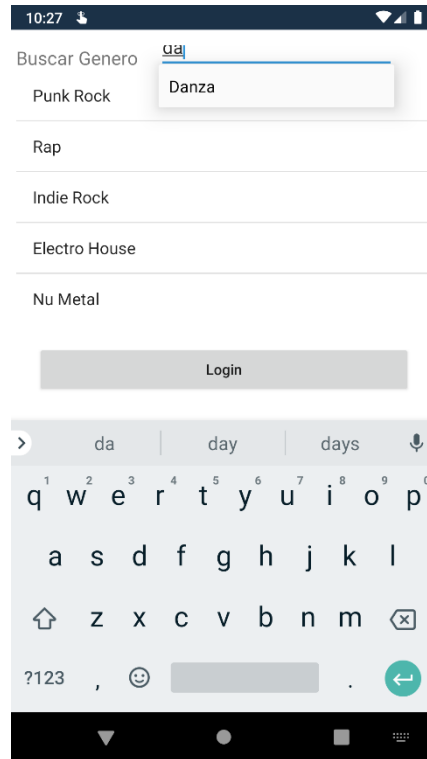


Figura 5. Busqueda eventos App CulturalCities.



Figura 6. Registro usuario App CulturalCities.

7.6. Despliegue del sistema

El despliegue del sistema se hace 100% en la nube..

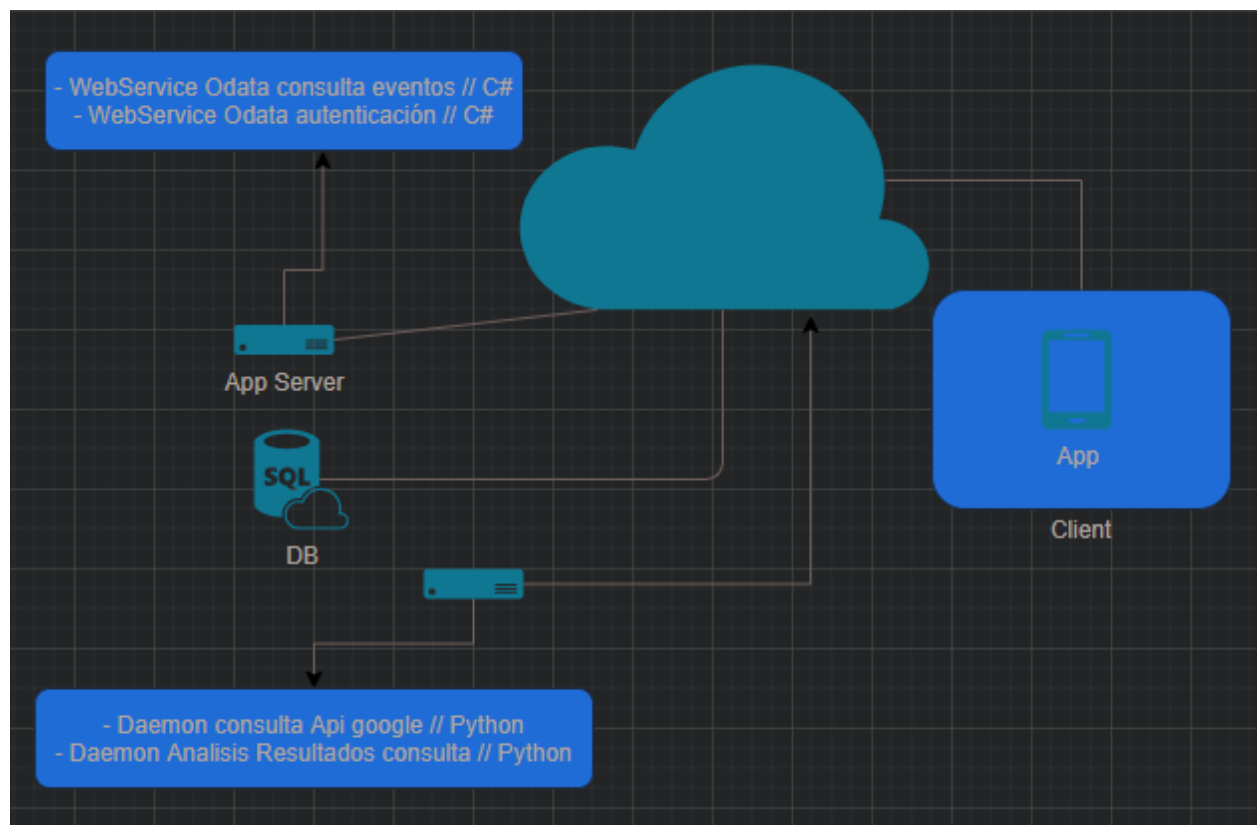


Figura 7. Despliegue del sistema.

7.6.1. Manual de usuario

Ver Anexo 1

7.6.2. Manual de instalación

- Creación de instancia en SQL Azure. Ver Anexo 2.
- Creación de instancia en App Service. Ver Anexo 3
- Instalación módulos necesarios para la ejecución de Cultural Bot. Ver Anexo 4
- Programación de tareas para la ejecución periódica de Cultural Bot. Ver Anexo 5

8. CONCLUSIONES

- Este proyecto nos muestra la importancia del trabajo en equipo, lo cual permite el desarrollo de módulos en paralelo, logrando de esta forma entregas funcionales en más corto tiempo; a menudo dentro de la mala planificación de nuestras actividades laborales y de la vida cotidiana, encontramos cuellos de botella al tener que esperar que alguien culmine una actividad que otra persona debe continuar, con una buena planificación se puede lograr hacer dos actividades del mismo proyecto a tiempo o diferentes proyectos, mientras se termina un entregable sobre el cual el desarrollador pueda continuar.
- La utilización de diferentes tecnologías en el desarrollo y puesta en marcha de Cultura Cities, permitió el crecimiento técnico del equipo, este enriquecimiento fue generado por cada uno de los problemas que se presentaron en la marcha y puesta a punto del proyecto y que genero espacios de debate los cuales llevaron a encontrar soluciones que antes el equipo no conocía.
- Sin embargo, deja grandes retos, ya que la implementación en una segunda fase del proyecto, trae un alto flujo de información y la tarea del equipo será procesar toda esta información en pro de encontrar las mejoras sugerencias al usuario sin ser invasivos con la publicidad.

REFERENCIAS

- Theory, G. (s. f.). PROGRAMAR TAREAS EN LINUX USANDO CRONTAB. Recuperado 4 de junio de 2020, de <https://geekytheory.com/programar-tareas-en-linux-usando-crontab>.
- Daniel Rodríguez. (2018, 26 octubre). Archivos JSON con Python: lectura y escritura. Recuperado 25 de mayo de 2020, de <https://www.analyticslane.com/2018/07/16/archivos-json-con-python/>
- D. (2018, 6 julio). Paso 1: Configuración del entorno de Python de pyodbc - SQL Server. Recuperado 1 de junio de 2020, de <https://docs.microsoft.com/es-es/sql/connect/python/pyodbc/step-1-configure-development-environment-for-pyodbc-python-development?view=sql-server-ver15>
- Calculadora de precios. (s. f.). Recuperado 10 de junio de 2020, de <https://azure.microsoft.com/es-mx/pricing/calculator/>
- Pricing · Plans for every developer. (s. f.). Recuperado 11 de junio de 2020, de <https://github.com/pricing>
- How to Use Google Custom Search Engine API in Python - Python Code. (s. f.). Recuperado 20 de mayo de 2020, de <https://www.thepythoncode.com/article/use-google-custom-search-engine-api-in-python>
- M. (2019, 1 julio). 5.1 Basic authentication over HTTPS - OData. Recuperado 2 de mayo de 2020, de <https://docs.microsoft.com/en-us/odata/webapi/basic-auth>
- XML API reference | Custom Search | . (s. f.). Recuperado 5 de mayo de 2020, de https://developers.google.com/custom-search/docs/xml_results#BooleanOrqt
- Custom Search API | . (s. f.). Recuperado 5 de mayo de 2020, de <https://developers.google.com/custom-search/v1/reference/rest/v1/Search>

ANEXOS

Anexo 1

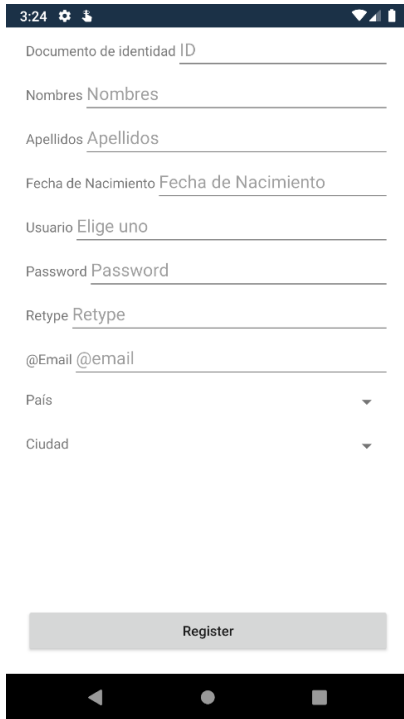
Manual de Usuario.

1. Abrir App Cultural Cities.



Figura 8. Inicio App Culturl Cities

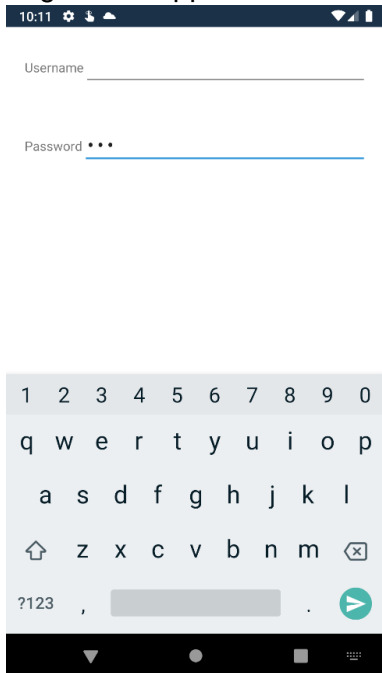
2. Registrar datos de usuario.



A screenshot of a mobile application's registration screen. At the top, the status bar shows the time 3:24 and various icons. The form consists of several input fields: 'Documento de identidad ID', 'Nombres Nombres', 'Apellidos Apellidos', 'Fecha de Nacimiento Fecha de Nacimiento', 'Usuario Elige uno', 'Password Password', 'Retype Retype', and '@Email @email'. Below these are two dropdown menus for 'País' and 'Ciudad'. A 'Register' button is positioned at the bottom of the form. The Android navigation bar is visible at the very bottom.

Figura 9. Registro usuario App CulturalCities.

3. Logueo en App Cultural Cities.



A screenshot of a mobile application's login screen. The status bar at the top shows the time 10:11. The form has two input fields: 'Username' and 'Password'. The password field is currently obscured by three dots. Below the form, a standard QWERTY keyboard is displayed, indicating that the user is in the process of typing. The Android navigation bar is visible at the bottom.

Figura 10. Logueo App CulturalCities

4. Lista de eventos en App Cultural Cities.

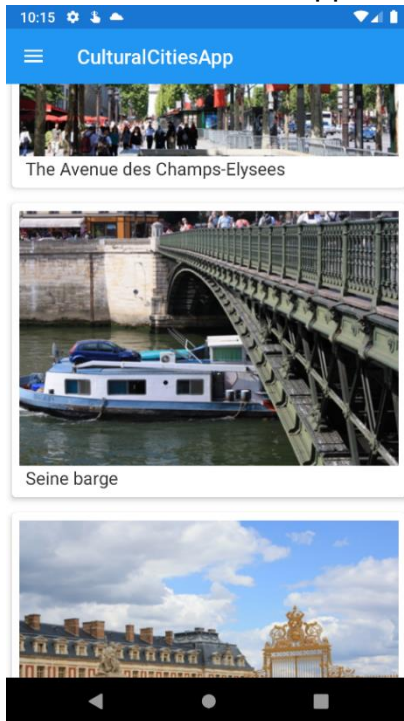


Figura 11. Lista eventos App CulturalCities.

5. Búsqueda de eventos en App Cultural Cities

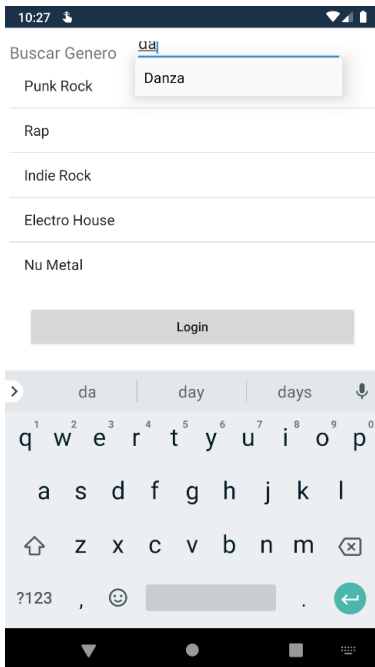


Figura 12. Búsqueda eventos App CulturalCities.

Anexo 2

Creación de instancia en SQL Azure.

1. Acceder al portal de Azure
2. En el panel de control ingresar a SQL Databases.



Figura 13. Panel de Control Azure.

3. Luego se da click en agregar.

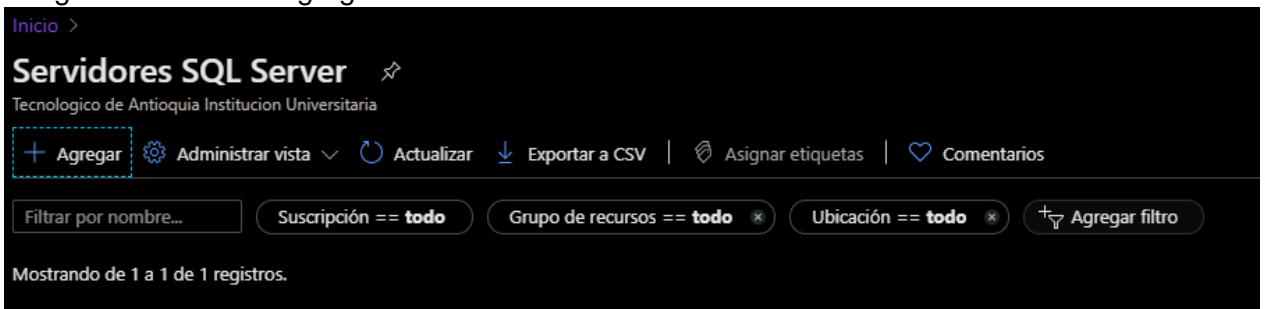


Figura 14. Panel de Control SQL Azure-1.

4. Se ingresa la información solicitada.

[Inicio](#) > [Servidores SQL Server](#) >

Crear un servidor de SQL Database



Microsoft



Básico Redes Configuración adicional Etiquetas Revisar y crear

Un servidor de SQL Database es un contenedor lógico para la administración de bases de datos y grupos elásticos. Complete la pestaña **Básico**; después, vaya a **Revisar y Crear** para aprovisionar los valores predeterminados inteligentes, o bien visite cada pestaña para personalizarla. [Más información](#) 

Detalles del proyecto

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.


Suscripción *  

Grupo de recursos *  
[Crear nuevo](#)

Detalles del servidor

Especifique la configuración necesaria para este servidor, incluidos un nombre y la ubicación.

Nombre del servidor * .database.windows.net

Ubicación * 

Cuenta de administrador

Inicio de sesión del administrador del

Revisar y crear **Siguiente: Redes >**

Figura 15. Formulario Creación SQL Database.

5. Se ingresa a la instancia creada.

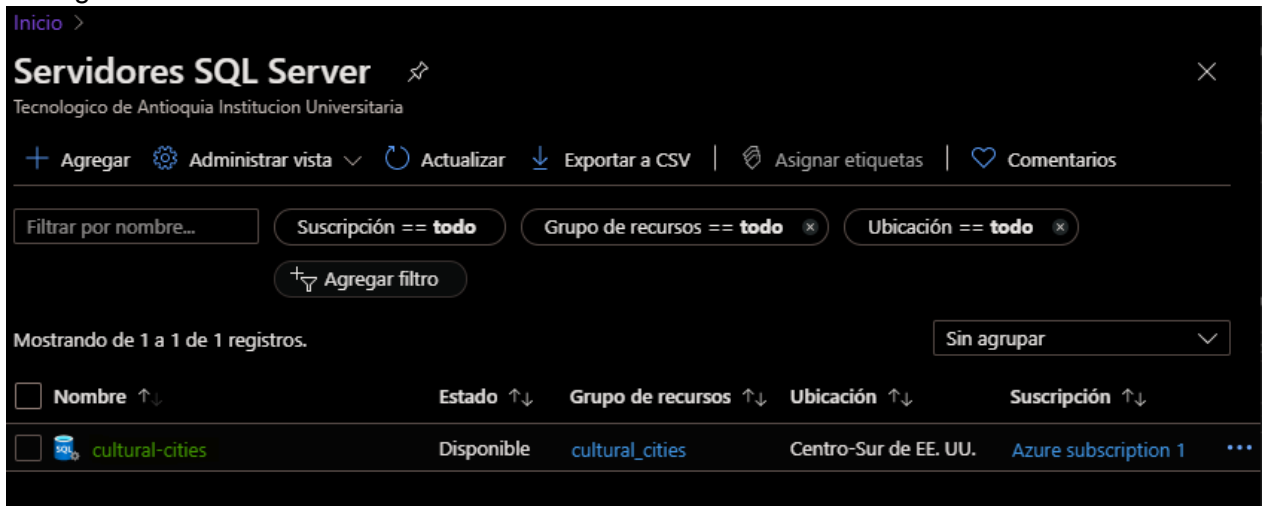


Figura 16. Panel de Control SQL Azure-2.



Figura 17. Panel de Control SQL Azure-3.

6. Y luego se ubica en el editor de consultas y click en abrir consulta para subir el script de la base de datos.

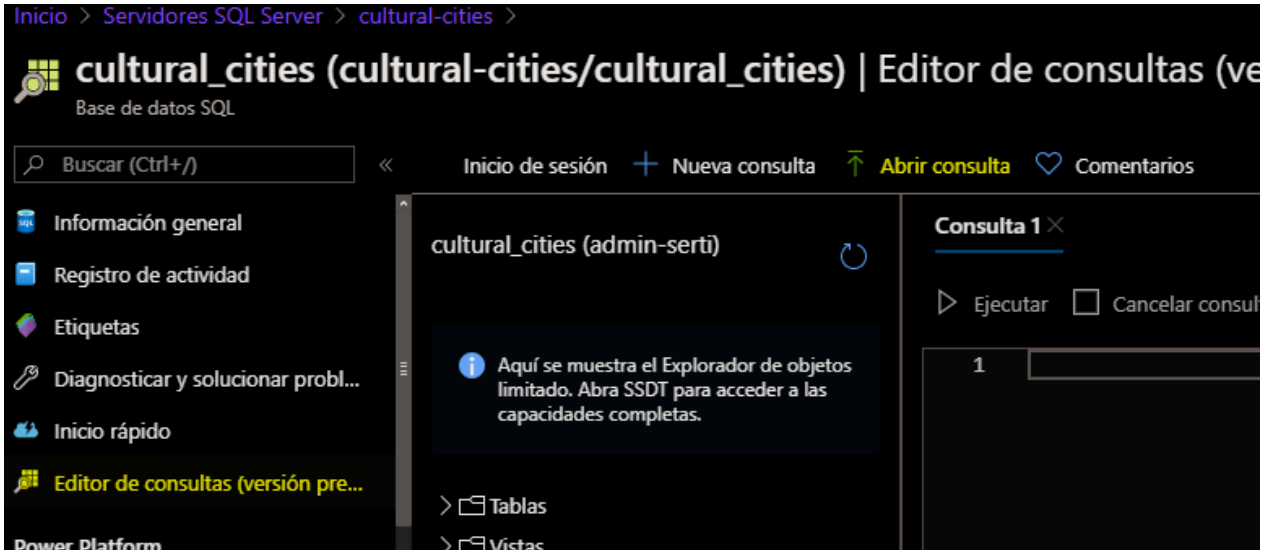


Figura 18. Editor Consultas SQL Database.

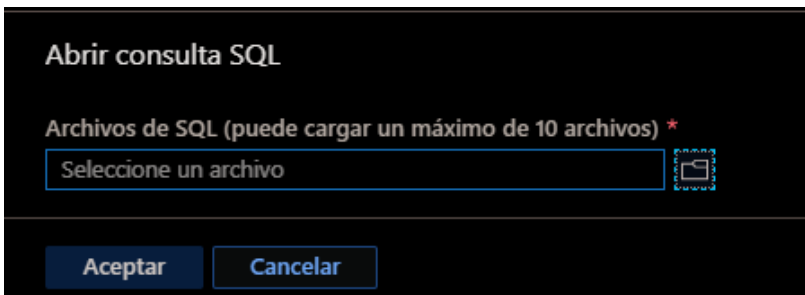


Figura 19. Editor Consultas SQL Database Subir Archivo-1.

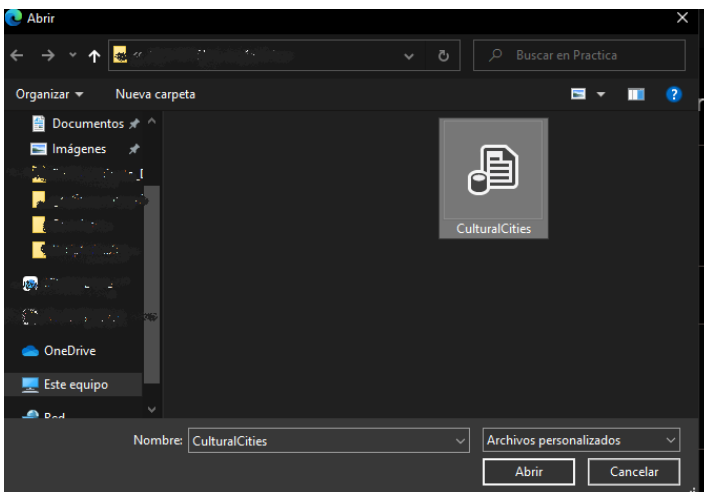


Figura 20. Editor Consultas SQL Database Subir Archivo-2.

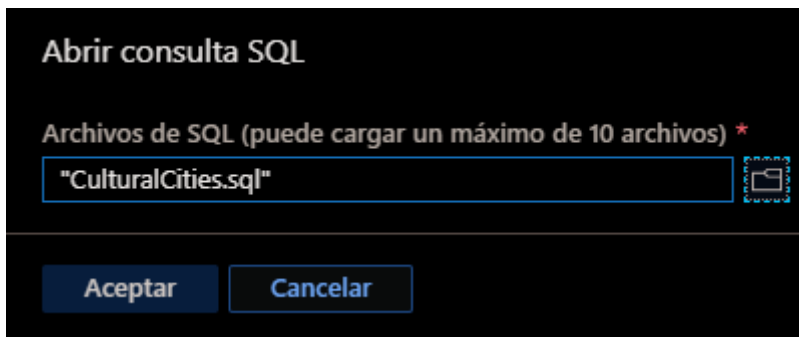


Figura 21. Editor Consultas SQL Database Subir Archivo-3.

7. Y ya podemos disfrutar de la base de datos

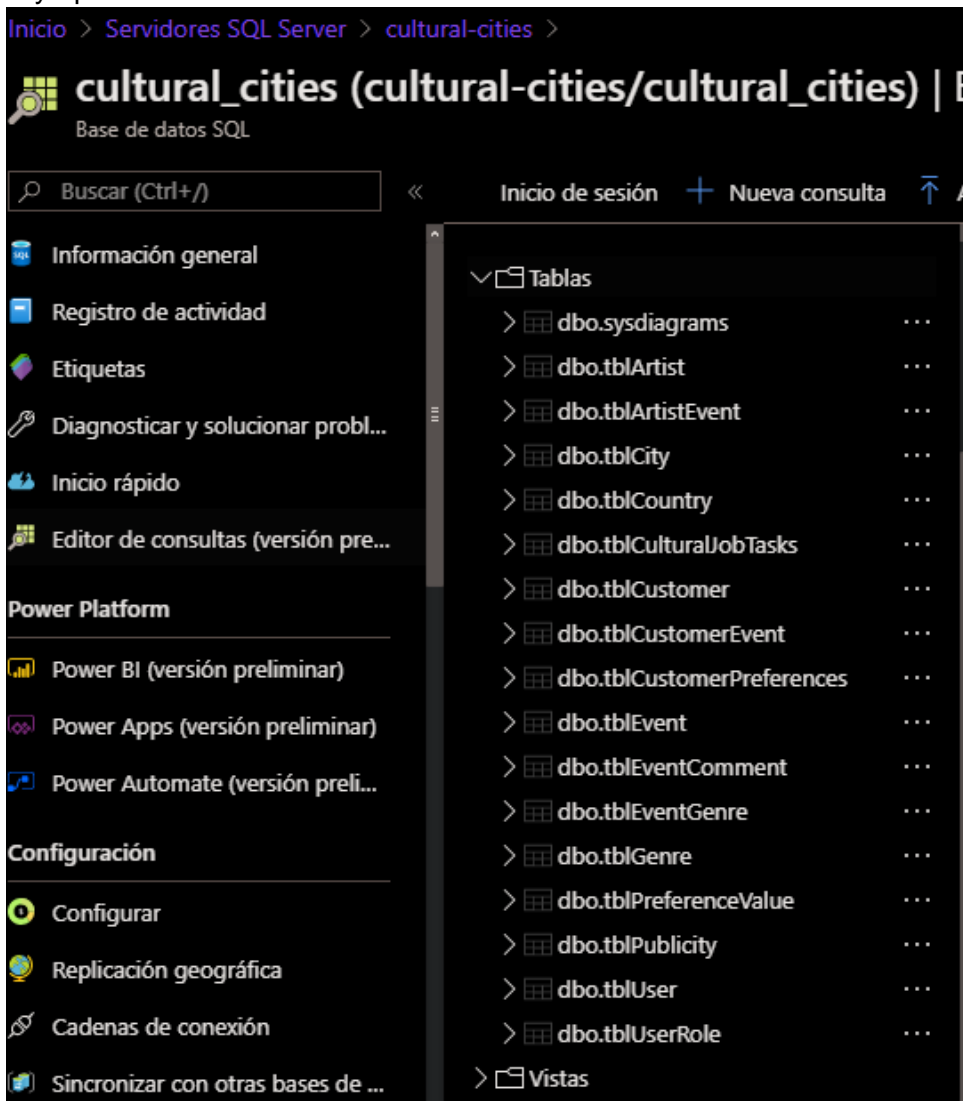


Figura 22. Tablas Base de datos.

Anexo 3.

Creación de instancia en App Service.

Se requiere visual studio 2019

1. Abrimos visual studio 2019 y abrimos el proyecto.

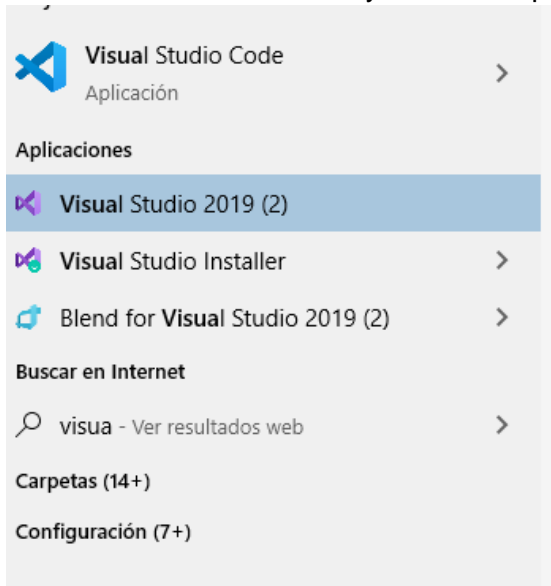


Figura 23. Acceso a Visual Studio 2019.

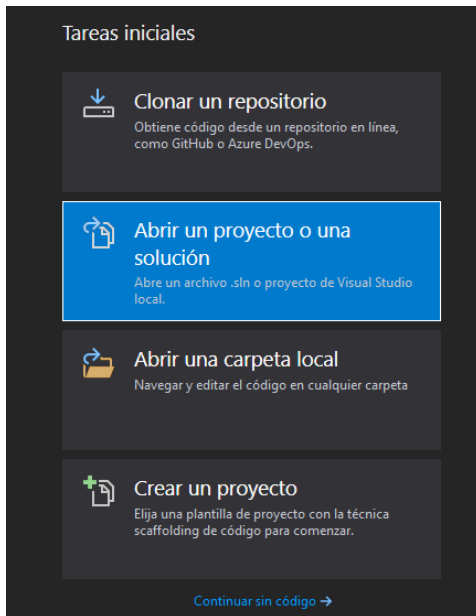


Figura 24. Abrir proyecto

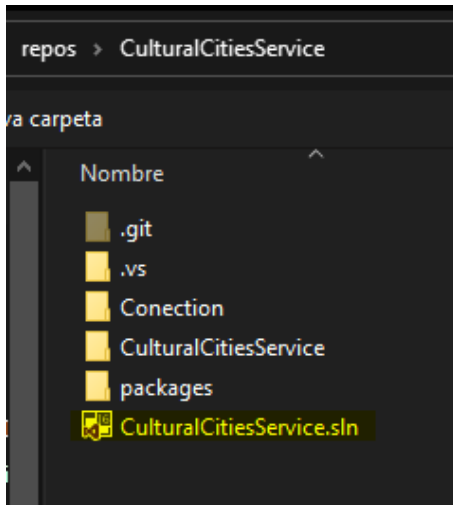


Figura 25. Seleccionar proyecto.

2. Luego de cargado el proyecto se procede a modificar el script de conexión con los datos de la base de datos creada previamente.

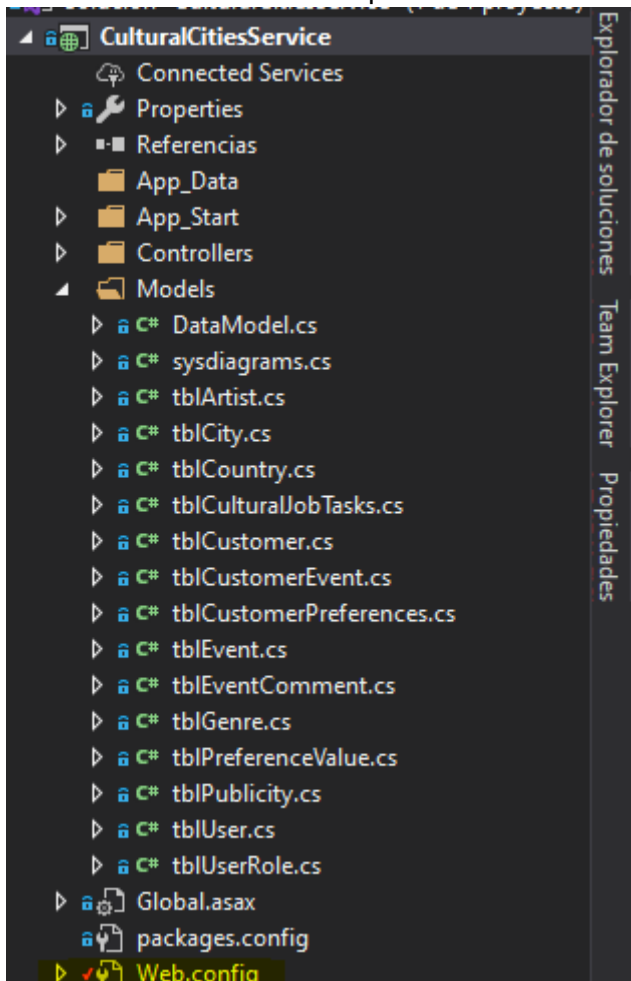


Figura 26. Abrir Archivo de configuración.

```
<connectionStrings>
  <add name="CultCitiesDataModel" connectionString="data source=tcp:cultural-cities.database.windows.net,1433;
    initial catalog=cultural_cities;persist security info=True;user id= USER ;password= PASSWORD ;
    MultipleActiveResultSets=True;App=EntityFramework;" providerName="System.Data.SqlClient" />
</connectionStrings>
```

Figura 27. Modificar String de conexión.

3. Luego se publica la aplicación en Azure. En Explorador de soluciones, haga clic con el botón derecho en el proyecto CulturalCitiesService y seleccione Publicar.

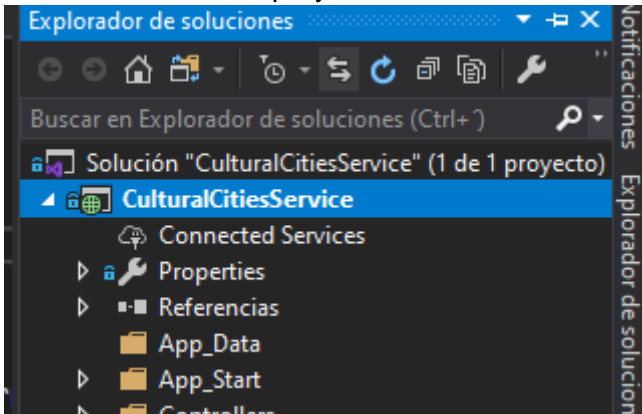


Figura 28. publicación en Azure-1.

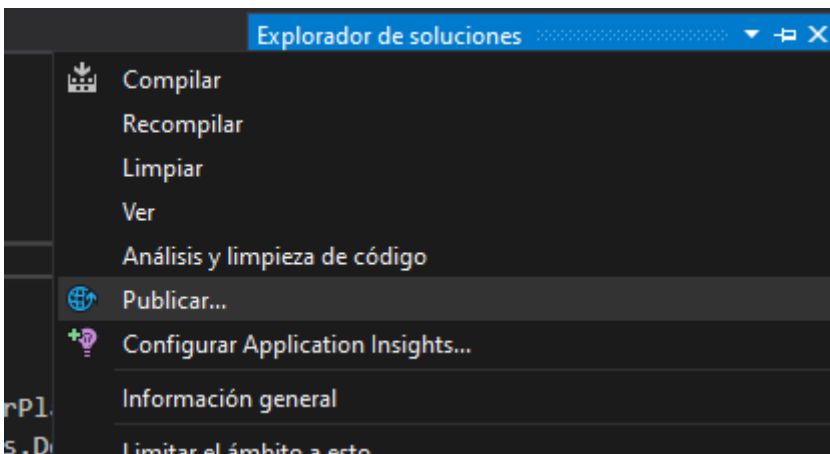


Figura 29. publicación en Azure-2.

4. Elija publicar y publique automáticamente en Azure.

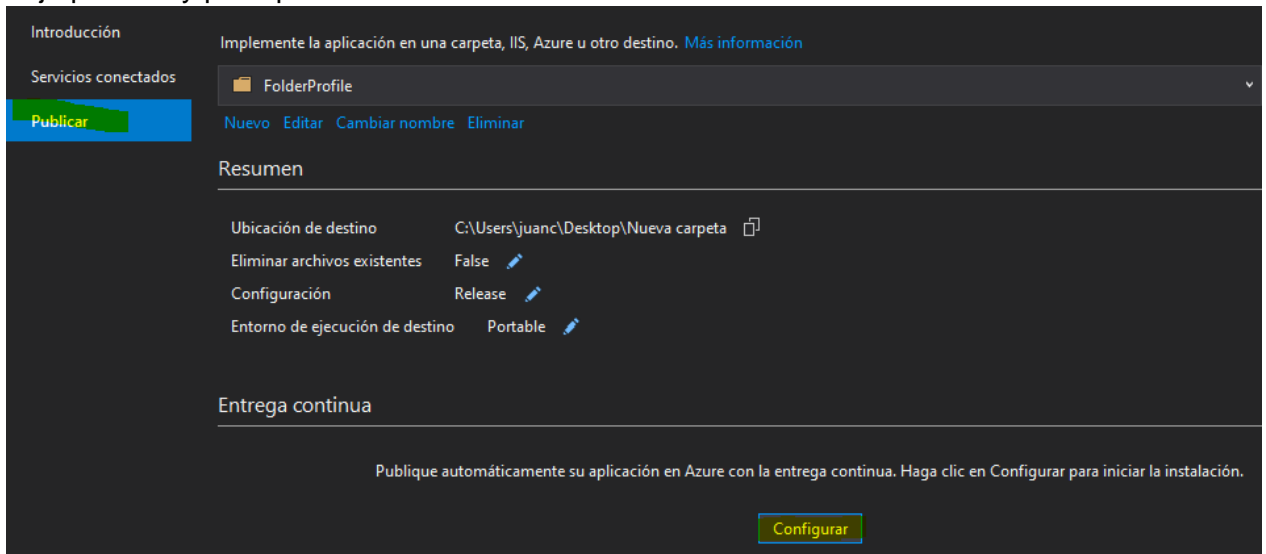


Figura 30. Publicación en Azure-3.

5. Loguese en su cuenta de Azure e Ingrese la información solicitada y edite los datos de la App Service.

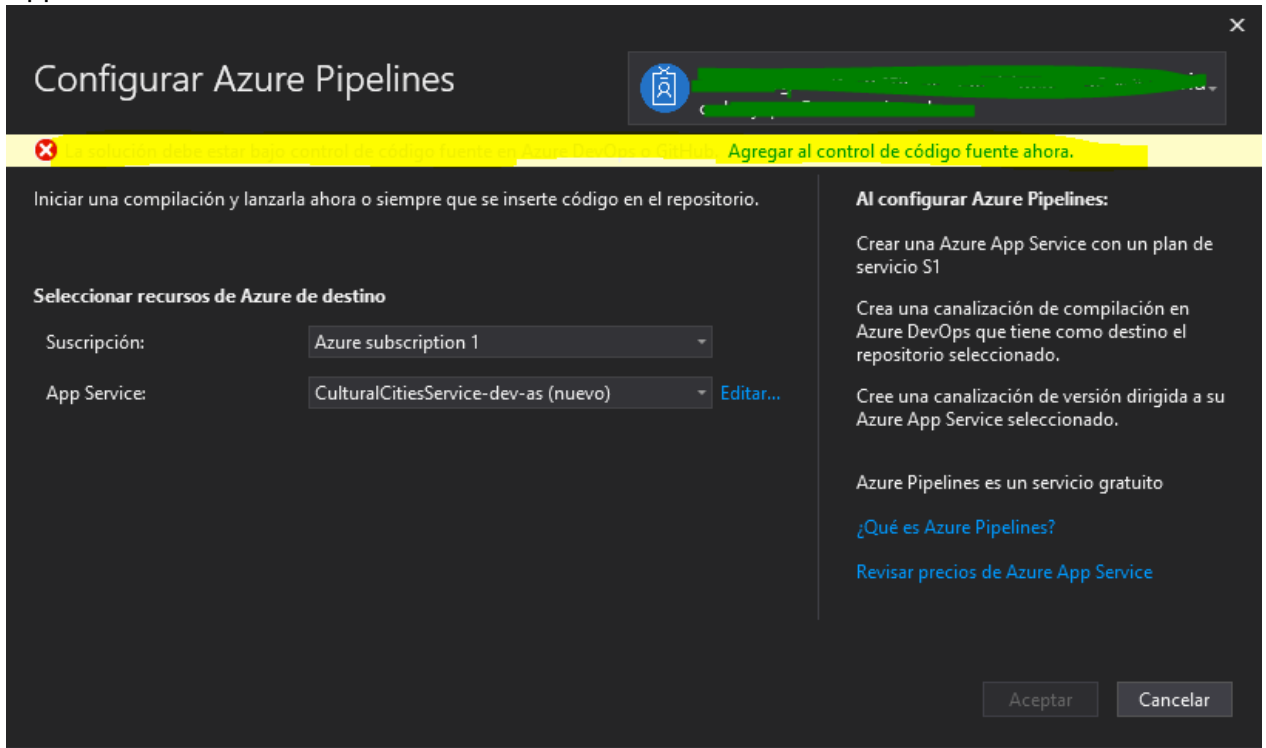


Figura 31. Configuración servicio Azure-1.

Click en agregar control de código fuente y pedirá conectar con Github o con Azure DevOps, en este caso publicamos en Azure DevOps y editamos App Service para seleccionar el plan del servidor

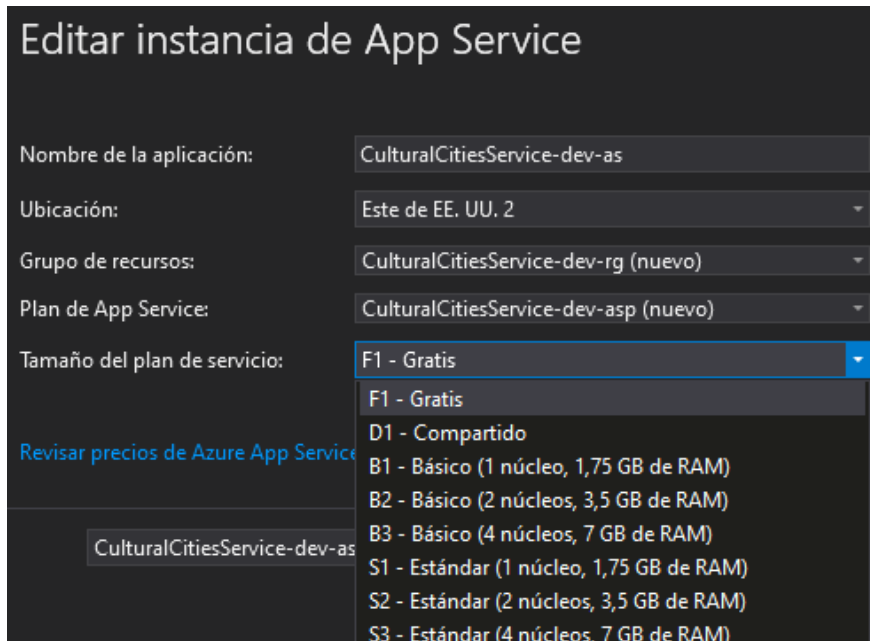


Figura 32. Configuración servicio Azure-2.

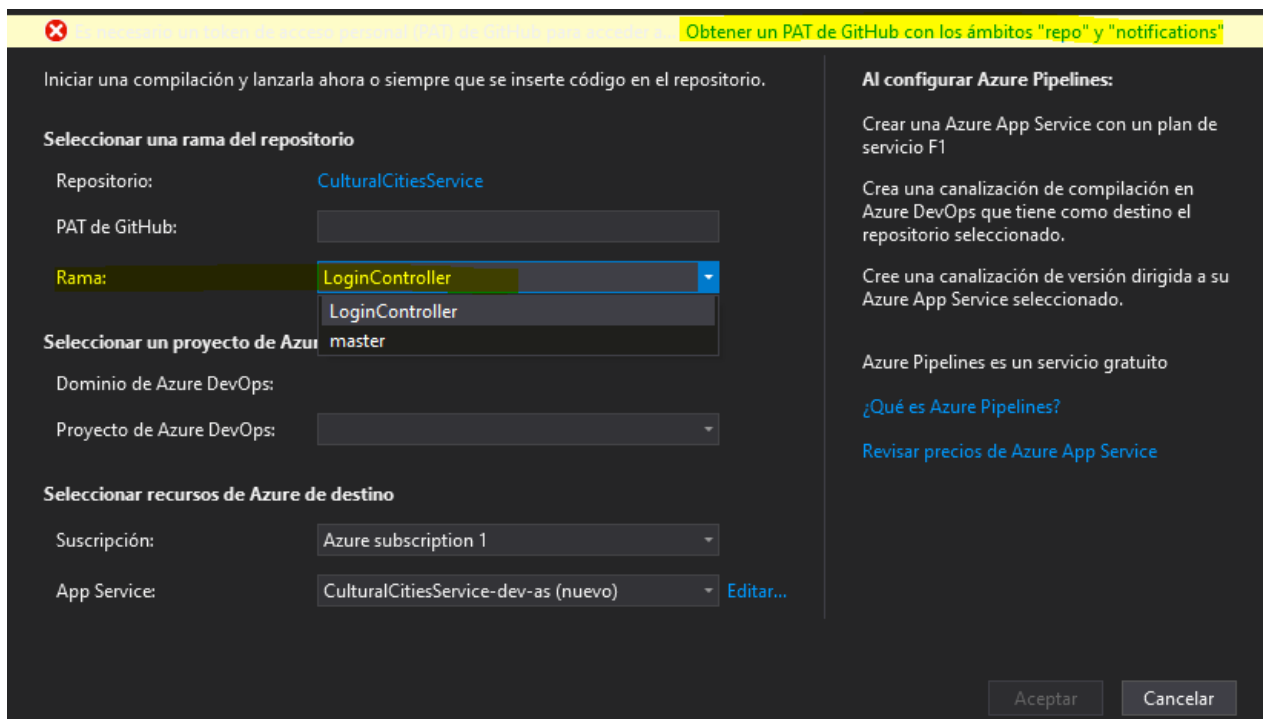


Figura 33. Configuración servicio Azure-3.

Luego Aceptar y esperamos hasta que el proyecto se cree en la nube

```
Azure App Service: http://CulturalCitiesService-dev-as.azurewebsites.net  
Proyecto de Azure DevOps: CulturalCitiesService (https://dev.azure.com/culturalcities/CulturalCitiesService)  
Canalización de compilación de Azure DevOps: https://dev.azure.com/culturalcities/CulturalCitiesService/\_build?definitionId=1  
Canalización de versión de Azure DevOps: https://dev.azure.com/culturalcities/CulturalCitiesService/\_releases?definitionId=1
```

Figura 34. Configuración servicio Azure-4.

Anexo 4

Instalación módulos necesarios para la ejecución de Cultural Bot en Ubuntu Linux.

1. Abra la terminal de comandos linux e instale el controlador Python para SQL (pyodbc) con el administrador de paquetes de **pip - Python**
pip install pyodbc
2. Instale la librería de Google para la ejecución de CSE
pip install --upgrade Google-api-python-client
3. Copie los archivos de Cultural Bot en el servidor linux

Anexo 5

Programación de tareas para la ejecución periódica de Cultural Bot en Ubuntu Linux

1. Editar contrab para la creación de la tarea.
sudo crontab -e
2. Anexe las siguientes líneas para la ejecución cada 6 horas de Cultural Bot.
* 0 * * * /rutaDelProyecto/CulturalCities/culturalBot.py
* 6 * * * /rutaDelProyecto/CulturalCities/culturalBot.py
* 12 * * * /rutaDelProyecto/CulturalCities/culturalBot.py
* 18 * * * /rutaDelProyecto/CulturalCities/culturalBot.py