



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DESARROLLO DE UN SOFTWARE DE GESTIÓN DE PROCESOS
INTERNOS DE OTECS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

NICOLÁS DANIEL JARA ROJAS

PROFESORA GUÍA:
SERGIO F. OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:
PABLO BARCELO BAEZA
PABLO GONZÁLEZ JURE

SANTIAGO DE CHILE
2018

RESUMEN DE LA MEMORIA PARA OPTAR AL
TÍTULO DE: Ingeniero Civil en Computación
POR: Nicolás Daniel Jara Rojas
FECHA: septiembre de 2018
PROFESOR GUÍA: Sergio F. Ochoa Delorenzi

DESARROLLO DE UN SOFTWARE DE GESTIÓN DE PROCESOS INTERNOS DE OTECS

El presente documento describe la implementación de un sistema web bajo la arquitectura de aplicación de página única (en inglés, SPA: Single Page Application), con el objetivo de dar apoyo a la gestión interna de OTECs (Organismos Técnicos de Capacitación), dando una experiencia más fluida a los usuarios del sistema a través de una aplicación de escritorio.

Las tecnologías escogidas en este proyecto permiten generar una arquitectura modular, que facilita la modificación de su estructura, con el objetivo de agregar o quitar servicios según los procesos internos que posea un OTEC en particular. Además, las funcionalidades desarrolladas en este sistema permiten apoyar procesos transversales de un OTEC, como por ejemplo: registrar usuarios, crear y corregir pruebas, y realizar encuestas de satisfacción, entre otras.

Este sistema web permite al OTEC, registrar diversos tipos de usuario: internos, empresas y estudiantes regulares. Entre las principales funciones de los usuarios internos (empleados del OTEC), se encuentra: registrar estudiantes y empresas, generar y revisar pruebas para las capacitaciones, cargar documentos de apoyo, y generar credenciales, diplomas y actas de entrega. Por su parte, los usuarios empresa pueden: controlar el avance los alumnos, y descargar las credenciales, diplomas y actas de entrega. Finalmente, estudiantes regulares pueden ver su participación en las capacitaciones, y descargar sus credenciales, diplomas y responder pruebas.

La solución desarrollada permite a la OTEC reducir el número de errores en la carga de información, y evitar la duplicidad de la misma. Por otro lado, al ser éste un sistema online, permite el trabajo en paralelo y distribuido, siempre que se cuente con acceso a internet.

Este trabajo de memoria se desarrolló en el marco un OTEC particular, quien destacó el software como un elemento diferenciador respecto a otros OTEC. Aunque este proyecto será continuado con el objetivo de cubrir todas las necesidades de un OTEC, los resultados actuales del software han sido evaluados positivamente, y le permiten al OTEC operar con niveles de error muy por debajo de sus niveles históricos.

Dedicado a mi mamá y a mi tata, y mis tres hermanitas, por la constante preocupación, cariño y apoyo brindado durante toda mi vida. A Cony por todo su apoyo para sacar este proyecto adelante. Y a mi profesor guía quién hizo de esta última etapa una muy agradable y nutritiva experiencia.

AGRADECIMIENTOS

Tras este largo camino emprendido de formación universitaria, formación laboral y trabajo de memoria, muchas personas han estado apoyándome y son a las cuales me gustaría agradecer.

A Daniel Jara mi tata, quién a sido un ejemplo de esfuerzo y de persona a seguir.

A Alejandra Rojas mi mami, quién a sido un ejemplo de cariño y amor incondicional.

A mi hermanas, Daniela, Paula y Michel, quienes siempre han estado acompañándome en este proceso.

A Cony, por su apoyo y cariño incondicional durante todo este proceso universitario.

A Javiera por su compañía y risas que siempre son bienvenidas.

A María Paulina por su seriedad y cariño.

A mi tío Samy, quién me apoyó en mi travesía del campo a Santiago.

A mis amigos y familiares que han aportado con su granito de arena.

A mi profesor guía, quién confió en mi proyecto y me apoyó en su realización.

A mi compa quién estuvo codo a codo apoyando en el trabajo.

A mis abuelitos Toño, Pelao y mis abuelitas Esmeralda, Nana y Nancy, quiénes siempre se preocupan de mi bienestar.

Finalmente, quiero agradecer a mi abuelito Fernando quién está cuidándome donde sea que esté, y con seguridad sé que está con su vasito de chicha celebrando como corresponde.

Muchas Gracias a todos.

TABLA DE CONTENIDO

1	Introducción	1
1.1	<i>Aspectos generales</i>	1
1.2	<i>Justificación y motivación</i>	2
1.3	<i>Posibles soluciones</i>	3
1.4	<i>Objetivos de la memoria</i>	5
1.5	<i>Estructura del documento de memoria</i>	5
2	Marco Teórico	7
2.1	<i>Los OTEC y su rol</i>	7
2.2	<i>La nueva Ley Regulatoria de los OTEC y sus consecuencias</i>	8
2.3	<i>Aspectos generales de solución requerida</i>	9
2.4	<i>Análisis de potenciales soluciones</i>	10
3	Concepción de la Solución	12
3.1	<i>Identificación de los usuarios de la plataforma</i>	12
3.2	<i>Principales casos de uso de la solución</i>	13
3.3	<i>Caso de uso primarios</i>	14
3.4	<i>Casos de uso secundarios</i>	16
3.5	<i>Procesos a apoyar</i>	18
3.6	<i>Tecnologías escogidas</i>	22
3.7	<i>Diseño de la arquitectura</i>	24
3.8	<i>Modelo de datos</i>	27
3.9	<i>Componentes de negocio</i>	29
3.10	<i>estructura del back-end</i>	34
3.11	<i>API implementada</i>	36
4	Interfaces del sistema	39
4.1	<i>Interfaces del Usuario Administrador</i>	41
4.2	<i>Interfaces del Usuario Relator</i>	44
4.3	<i>Interfaces del Usuario Estudiante</i>	45
4.4	<i>Interfaces del Usuario Empresa</i>	46
5	Evaluación de la solución	48
6	Conclusiones y trabajo a futuro	49
7	Bibliografía	51

8	Anexos	52
8.A	<i>Manual de instalación.....</i>	<i>52</i>
8.A.1	<i>Virtualenv y VirtualEnvWrapper</i>	<i>52</i>
8.A.2	<i>Crear ambiente para el proyecto Django back-end y front-end.....</i>	<i>53</i>
8.A.3	<i>Instalación y configuración de uWSGI.....</i>	<i>54</i>
8.A.4	<i>Automatizar la ejecución de uWSGI.....</i>	<i>55</i>
8.A.5	<i>Instalación y configuración de Nginx como proxy inverso.....</i>	<i>56</i>
8.A.6	<i>Modificaciones a los proyectos back-end y front-end.</i>	<i>58</i>
8.B	<i>Documentación API proyecto back-end.</i>	<i>59</i>

ÍNDICE DE FIGURAS

Figura 1. Arquitectura física de la solución.....	9
Figura 2. Diagrama casos de usos y sus actores.....	14
Figura 3. Proceso realización de una capacitación en el sistema web.....	19
Figura 4. Proceso registro de alumnos para un curso.	20
Figura 5. Proceso generación de credenciales.	20
Figura 6. Proceso de publicación de material de apoyo.....	21
Figura 7. Proceso de solicitud de credenciales.	21
Figura 8. Arquitectura de la solución.....	25
Figura 9. Diagrama Entidad Relación de la Base de Datos del sistema.	27
Figura 10. Módulo Administración y sus submódulos.	29
Figura 11. Módulo Empresa y sus submódulos.....	31
Figura 12. Módulo Estudiante y sus submódulos.	32
Figura 13. Módulo Relator y sus submódulos.....	33
Figura 14. Diagrama Funcionamiento Back-end.	35
Figura 15. Mapa de navegación de la aplicación web para el perfil Administrador.	39
Figura 16. Mapa de navegación de la aplicación web para el perfil Relator.	40
Figura 17. Mapa de navegación de la aplicación web para el perfil Estudiante.	40
Figura 18. Mapa de navegación de la aplicación web para el perfil Empresa.	41
Figura 19. Vista lista de empresas.	42
Figura 20. Formulario de registro de empresa.....	42
Figura 21. Vista sección cursos.....	43
Figura 22. Formulario para agregar alumnos a un curso.....	44
Figura 23. Formulario para crear una prueba y sus preguntas.....	45
Figura 24. Vista de cursos en los que participa el estudiante.	46
Figura 25. Lista de estudiantes pertenecientes a la empresa.	47

1 INTRODUCCIÓN

1.1 ASPECTOS GENERALES

En Chile existen 2.673 Organismos Técnicos de Capacitación (OTEC) vigentes al 19 de junio del 2018, que adhieren a la caracterización establecida en la Ley 19.518 y Ley 19.967, y han sido acreditados por el SENCE (Servicio Nacional de Capacitación y Empleo)¹. A pesar de que el Gobierno regula el funcionamiento de estas instituciones y define un conjunto de instrumentos internos con los que los OTEC deben contar en su operación diaria (por ejemplo: libro de clases, registro de notas, etc.), el Estado no provee ningún mecanismo o soporte computacional a nivel de software, que les ayude a implementar estos instrumentos y a operar según lo que establece la normativa actual.

Dado esto, cada OTEC maneja de forma independiente y particular sus procesos internos a la hora de brindar servicios e implementar (según su mejor entendimiento) los instrumentos que les exige la norma, entre los cuales están los siguientes:

1. *Libro de clases*: Este instrumento permite especificar la lista de alumnos que conformarán una capacitación. Además, en este libro se lleva registro de asistencia a clases, evaluaciones, notas de los alumnos y anotaciones varias.
2. *Encuesta de satisfacción del participante*: Este instrumento permite realizar la encuesta durante y posterior a la capacitación brindada. A través de esta encuesta se busca generar retroalimentación del servicio entregado por el OTEC.

¹ <http://www.sence.cl/portal/Perfil/Organismos/OTEC/>

3. *Acta de aprobación*: Este instrumento permite registrar el acta del curso, el detalle de las evaluaciones de los alumnos, el porcentaje de asistencia y su condición de aprobado o reprobado.
4. *Rendición de gastos*: Este instrumento que permite al OTEC gestionar los gastos realizados en la capacitación, junto con el detalle de las boletas, facturas y documentos de respaldo de gastos, llevando un registro claro y ordenado de estos.
5. *Generación de diplomas*: Este instrumento permite generar los diplomas de manera automática para los alumnos aprobados en la capacitación brindada.
6. *Generación de credenciales*: Este instrumento permite generar las credenciales para los alumnos participantes de la capacitación. Esta credencial permite tener un acceso rápido a la documentación del alumno, como así también a los certificados de capacitación que éste posee. Además, esta credencial es utilizada como medio de comprobación de las capacitaciones adquiridas por el alumno.

Tras la búsqueda de software que permita apoyar los procesos internos de un OTEC, para implementar los instrumentos exigidos por la norma actual, se encontraron unos pocos, los cuales eran muy antiguos. Debido a eso, muchas de las tecnologías actuales y necesidades de los OTEC no estaban cubiertas, por lo tanto, muchos OTEC utilizan plantillas EXCEL como apoyo a sus procesos internos. Esta forma artesanal de gestionar los procesos internos y externos de una organización conlleva a diversos problemas, como, por ejemplo; duplicidad de la información, limitaciones de acceso a los datos, poca o nula seguridad, errores en el ingreso y/o manipulación de los datos, y dependencia de un computador particular para acceder y respaldar la información, entre otros. Debido a esto, se desarrolló un software online que cubre todos los procesos internos de un OTEC, según lo exigido por la norma actual, brindando un servicio más amplio, facilitando el trabajo interno y el manejo de la información del OTEC para con sus clientes.

1.2 JUSTIFICACIÓN Y MOTIVACIÓN

La motivación de desarrollar este software nació como una forma de brindar una herramienta para la gestión interna y externa de los OTEC, lo que genera valor agregado tanto para el OTEC, como para los clientes finales de éste. Con

el fin de dimensionar el tamaño de este desafío, se llevó a cabo un relevamiento en dos OTECs. Como resultado de ello, se identificaron 26 funcionalidades que debían ser desarrolladas para poder cumplir con el apoyo requerido y lo establecido por la normativa actual. El desarrollo de estas funcionalidades conlleva importantes desafíos en el ámbito de la ingeniería de software, los cuales deben ser superados a fin de tener un proyecto satisfactorio y llamativo para los OTEC. Entre los principales desafíos abordados están los siguientes:

1. *Diseño de una arquitectura modular*: Debido a las 26 funcionalidades identificadas para cubrir, tanto los procesos internos como externos de los OTEC, contar con una arquitectura de software apropiada se vuelve mandatorio, ya que el sistema debe poder agregar/modificar su funcionalidad de manera fácil, y también evolucionar a través de distintas tecnologías. El tiempo de vida y el esfuerzo de actualización de la solución desarrollada dependerá en gran medida de cuán bien haya sido diseñada la arquitectura del sistema.
2. *Implementación incremental de la solución*: El software desarrollado es muy grande, por lo tanto, éste debía poder implementarse de manera incremental. Además, se llevó a cabo esta tarea involucrando tecnología actual y de bajo costo, asegurando la robustez de la solución y permitiendo la accesibilidad de la misma. La idea es que a futuro una gran cantidad de OTECs usen el sistema, y para ello se debe minimizar o eliminar las barreras de entrada para la utilización del software en distintos escenarios (o realidades).
3. *Diseño de una interfaz de usuario usable*: Otro gran desafío que tiene asociado el software, es que no basta con ofrecer los servicios o herramientas de software a los OTEC, sino que además éstas deben ser fáciles de aprender y de usar por parte de los usuarios finales. Cabe destacar que los usuarios pueden ser muy diversos, por lo que pesa una exigencia importante sobre la usabilidad de las interfaces del sistema.

1.3 POSIBLES SOLUCIONES

Dado el problema detectado en el apoyo a los procesos internos y externos de un OTEC, surgen diversas soluciones potenciales, como por ejemplo el desarrollo de una aplicación móvil y/o de escritorio que se instale por demanda

a través de una secuencia de pasos seguidos por un usuario. La aplicación podría hacer uso de Internet para acceder a un servidor remoto (para llevar a cabo el intercambio de datos), o bien podría también crearse una aplicación servidora que se instale en un computador del OTEC. En ese caso, cualquier actualización al software (ya sea como parche o extensión) requiere acceso físico al computador correspondiente y la iniciativa de un usuario para llevar a cabo dicha actualización. Este enfoque, aunque es viable, es usualmente engorroso tanto para el desarrollador como para el cliente.

Por otra parte, en el caso de la aplicación móvil, resulta poco intuitivo y difícil de brindar una interfaz lo suficientemente clara y fácil de usar, como para ingresar y gestionar grandes volúmenes de información, pudiendo esto aumentar la tasa de errores al ingresar dicha información al sistema.

Otra posible solución es implementar una aplicación Web que se instale en el servidor del OTEC, y que sea accesible por los dispositivos clientes utilizando un navegador (ya sea en el celular o en notebook). Este enfoque tiene menores restricciones que el anterior respecto al uso y actualización de las funcionalidades del sistema, sin embargo, requiere que el OTEC tenga un servidor que pueda ser utilizado con ese propósito. Además, debe asegurarse un importante nivel de uptime para el servidor, para el software que éste ejecuta y para el vínculo que permite conectarlo a Internet. Estas restricciones podrían ser no abordables por aquellos OTECs que son pequeños, o que operan en zonas rurales donde el acceso a Internet es limitado.

Finalmente, otra opción considerada fue el desarrollar un sistema web que funcione como un servicio disponible en la nube. Aunque este esquema tiene similitudes con el último presentado, en este caso se reducen considerablemente todos los riesgos asociados a la disponibilidad y ancho de banda de acceso al servidor, por lo que el usuario sólo necesita contar con un navegador y acceso a Internet para utilizar el sistema. Además, cada cambio que se genera al sistema, y el desarrollo de nuevas funcionalidades, pueden ser incorporadas a la aplicación de forma transparente y transversal a todos los clientes (OTECs), reduciendo así el costo y el tiempo asociado a las mantenciones y extensiones a la solución.

1.4 OBJETIVOS DE LA MEMORIA

El objetivo general de este trabajo de memoria es diseñar la arquitectura del software de una aplicación web, que permita a los OTEC gestionar los procesos internos y externos de acuerdo con lo que establece la normativa actual. Debido a que el tamaño de este software es muy grande (involucra al menos 26 funcionalidades como los enunciados en la sección 1.1), en esta memoria se implementa la arquitectura base y las siete funcionalidades identificadas en la sección 1.2. Por lo tanto, los objetivos específicos que se derivan del objetivo principal son los siguientes:

1. Desarrollar una arquitectura que permita una implementación gradual y modular del software de apoyo a la operación de los OTEC.
2. Desarrollar las siete funcionalidades identificadas en la sección 1.1, y de esa manera permitir al OTEC gestionar internamente la información respectiva del desarrollo de sus cursos de capacitación.

1.5 ESTRUCTURA DEL DOCUMENTO DE MEMORIA

La estructura utilizada en este documento consta de 6 capítulos, siendo estos los siguientes:

- *Capítulo 1: Introducción.* Se presentan los aspectos generales y motivación del tema, las posibles soluciones, los objetivos generales y específicos del trabajo de memoria.
- *Capítulo 2: Marco Teórico.* Corresponde a la revisión de la información del contexto en que se desenvuelve un OTEC. En este capítulo se muestran y explican los conceptos relacionados con el problema resuelto, y aquellos que sientan las bases para la resolución de dicho problema.
- *Capítulo 3: Concepción de la solución.* Corresponde a la revisión y explicación de la solución desarrollada. En este capítulo se muestra en detalle, los requisitos de la solución, los casos de uso de ésta, las tecnologías utilizadas, y el diseño de la solución concebida ante el problema abordado en este trabajo de memoria.

- *Capítulo 4: Interfaces del sistema.* Corresponde a las vistas con las que interactúan los usuarios del sistema. En este capítulo se muestra en detalle los principales flujos y sus interfaces con las que interactúa cada usuario del sistema desarrollado.
- *Capítulo 5: Evaluación de la solución.* En este capítulo se describe la evaluación realizada en una OTEC, y sus apreciaciones de dicho organismo respecto a la usabilidad y utilidad de la solución.
- *Capítulo 6: Conclusiones y trabajo a futuro.* Se señalan las conclusiones acerca del trabajo realizado, y se realizan propuestas de trabajo a futuro.

Finalmente se presenta la bibliografía utilizada y un anexo con material complementario, como, por ejemplo, la documentación de la API desarrollada, y la documentación de instalación, entre otros.

2 MARCO TEÓRICO

2.1 LOS OTEC Y SU ROL

El Servicio Nacional de Capacitación y Empleo (SENCE) nació al amparo del Decreto Ley 1.446 de 1976, el cual indica que las empresas tienen permitido deducir de impuestos el gasto efectuado en capacitación del personal de la empresa, en hasta el 1% de la planilla anual de remuneraciones. En este contexto, a nivel institucional del SENCE, la ley establece que el OTEC: “es toda persona jurídica que ejecuta acciones de capacitación”. Por otro lado, la Ley 19.825 estableció que los OTEC imparten cursos orientados a: “promover el desarrollo de las competencias laborales de los trabajadores, a fin de contribuir a un adecuado nivel de empleo, mejorar la productividad de los trabajadores y las empresas, así como la calidad de los procesos y productos”, esto bajo el objetivo del sistema de capacitación y empleo.

Al 10 de agosto de 2018 existen 2.797 OTECs vigentes, que adhieren a la caracterización establecida en las Leyes 19.518 y 19.967, y que han sido acreditados por el SENCE. En este marco, encontramos un estudio realizado por el Centro de Sistemas Públicos² del Departamento de Ingeniería Industrial de la Universidad de Chile, sobre la organización, estructura y funcionamiento del mercado de la capacitación en Chile en el 2015. Dicho estudio establece cuatro pilares en el modelo de negocios de los OTEC:

1. *Estructuración de la oferta.* Desarrollo de contenido para los cursos y definición del personal para su realización
2. *Formalización.* Inscripción de cursos inscritos y validados por el SENCE.
3. *Implementación.* Ejecución del curso.
4. *Promoción y ventas.* Mantención del vínculo comercial con la empresa-cliente.

² <http://www.sistemaspublicos.cl/>

Cabe destacar que en todos los pilares identificados en el modelo de negocio del OTEC, se desprende que el ciclo de vida de una capacitación se centra en el funcionamiento del día-a-día del OTEC.

2.2 LA NUEVA LEY REGULATORIA DE LOS OTEC Y SUS CONSECUENCIAS

Dentro de las regulaciones que existen para los OTEC, se encuentra la Norma Chilena de Calidad 2728:2015. Esta norma establece los requisitos para la implementación de un sistema de gestión de calidad, lo que les permitirá al OTEC contar con varias capacidades: a) disponer de información relevante para tomar decisiones estratégicas; b) orientar su gestión hacia la satisfacción del usuario y/o participante; c) generar periódicamente información de sus resultados financieros; y d) documentar su proceso de realización del servicio de capacitación, entre otros.

En el contexto de la norma chilena de calidad 2728, en su versión anterior 2013, se generaron una serie de modificaciones que los OTEC certificados bajo dicha norma, debieron realizar. Particularmente, estos organismos tuvieron que cambiar sus procesos internos para cumplir los nuevos requisitos establecidos por la Ley. Se dio un plazo de dos años para la incorporación de cambios, perdiendo la certificación en el caso que el OTEC no incorpore los nuevos procesos. Ejemplos de estos procesos son los siguientes: la generación de nueva documentación asociada a los cursos, y la incorporación de nuevos participantes en los flujos de capacitación. Además, este año (2018) se está solicitando que todo relator de una capacitación en particular debe poseer un contrato tipo, específico para dictar dicha capacitación. En caso contrario se genera una falta, pudiendo el OTEC ser sancionado. Por lo tanto, se hace necesario poder contar con una herramienta de software que apoye y facilite la adquisición de los nuevos procesos, dentro de la cultura y gestión interna del OTEC.

Respecto a los instrumentos que exige la norma actual para la operación de los OTEC, según la Resolución Exenta N° 5.477, donde se establece la obligatoriedad de uso del "Nuevo Libro de Clases Electrónico (LCE)", éste es el principal instrumento que considera el SENCE, y que es de uso obligatorio según lo indica la resolución exenta citada. El LCE es una aplicación de escritorio, que permite llevar un registro de asistencia de los participantes y relatores, durante las actividades de capacitación presencial. Ésta es una

aplicación provista por el SENCE, la cual tiene la desventaja de que no permite controlar otros procesos internos o externos del OTEC, por lo cual la herramienta no cumple con las necesidades requeridas por el proponente.

Considerando lo anterior, y como solución a los problemas detectados en la gestión interna y externa de los OTEC, se evidencia la necesidad de desarrollar una plataforma como la descrita en este trabajo de memoria. Esta plataforma no sólo busca ayudar a los OTEC a cumplir con las normas vigentes, sino también a facilitar sus procesos internos y con entidades gubernamentales.

2.3 ASPECTOS GENERALES DE SOLUCIÓN REQUERIDA

Como idea general de la solución a diseñar y desarrollar en esta memoria, cabe destacar que se trata de un Software como un Servicio (en inglés, SaaS: Software as a Service), el cual usualmente posee tres capas: el soporte de datos, la lógica de negocio y la interfaz de usuario.

Típicamente esta infraestructura de software se aloja en la nube, tal como se ejemplifica en la Figura 1. Arquitectura física de la solución. En ese escenario el usuario accede a dicha funcionalidad mediante un navegador Web, ya sea a través de computador o un smartphone, siendo sólo necesario una conexión a Internet.



Figura 1. Arquitectura física de la solución.

Desarrollar un software bajo la arquitectura antes mostrada, presenta varias ventajas, como, por ejemplo: una alta disponibilidad del sistema, bajo costo de actualización del software una vez que éste está en producción, y bajo costo de operación de la plataforma, entre otros. Además, este enfoque de operación

adhiera a la tendencia actual para ofrecer este tipo de soluciones a usuarios de manera masiva.

2.4 ANÁLISIS DE POTENCIALES SOLUCIONES

Tras la búsqueda de alternativas ya implementadas para dar solución al problema planteado, no se encontró ninguna solución tecnológica que contenga todos los módulos identificados para la gestión interna y externa del OTEC. Sin embargo, se encontraron algunas soluciones parciales al desafío planteado. Una de ellas es el sistema desarrollado por la empresa DATAWORK (empresa de facturación Electrónica). Esta empresa, junto a su nicho de negocio principal, ofrece una plataforma para la gestión parcial de los procesos de los OTECs.

Independientemente de la funcionalidad de dicha plataforma, DATAWORK entrega el código fuente del software a la OTEC, por lo cual éste último debe hacerse cargo de implantar y mantener dicho producto para poder utilizarlo. Esta actividad es usualmente muy difícil (sino imposible) de realizar por las OTEC, lo que lleva a que dichas organizaciones terminen usando herramientas genéricas como Office, para apoyar su gestión organizacional.

Algunas de las funciones que ofrece el software de DATAWORK son las siguientes: ingreso de clientes, generación de presupuestos de cursos, facturación electrónica y control de comisiones, entre otros. A pesar de que este software posee algunos módulos útiles para el OTEC, estos están enfocados netamente a la parte contable. Otro inconveniente de adoptar ese software es que la modalidad de pago del servicio sólo existe como venta de proyecto, y no como servicio mensual, lo cual es una barrera de entrada alta para los OTEC, debido al alto costo que esto significa.

Otro proponente de un software que también brindaría una solución parcial al problema planteado, es la empresa Haciendo Negocios S.p.A.³, la cual está orientada a prestar servicios de asesoría a pequeñas y medianas empresas con el objetivo de ayudar a crear un OTEC. Dentro de estas asesorías ellos potencian el uso de un software online para el apoyo a la gestión, el cual se encuentra en <http://academiahn.com/chamilo/>. Este sistema puede personalizarse para cada OTEC, pero aún el software se encuentra en desarrollo, por lo tanto, que no existe la posibilidad de considerarlo como una solución en el corto plazo a los problemas planteados en este trabajo de memoria.

Finalmente, encontramos los sistemas de planificación de recursos empresariales (en inglés, ERP: Enterprise Resource Planning). Estos permiten gestionar la automatización de procesos, disponibilidad de información en un solo punto central, entre otras cosas. La dificultad que presenta esta solución para el OTEC, es el alto costo, debido a la necesidad de personalizar el sistema. Otro punto importante de este tipo de soluciones es el alto impacto que tienen para poder instalarse dentro de una empresa, debido al hardware (tanto del cliente como del servidor) y a la infraestructura de redes que necesitan. Por lo tanto, estas soluciones delegan la responsabilidad de mantener infraestructura y hardware al OTEC; responsabilidad que muchas veces no puede ser asumida por organizaciones pequeñas, como se indica en la sección 1.3.

³ <http://www.tuotec.cl/nosotros>

3 CONCEPCIÓN DE LA SOLUCIÓN

En esta sección se presentan los principales componentes que forman parte de la solución descrita en este trabajo de memoria.

3.1 IDENTIFICACIÓN DE LOS USUARIOS DE LA PLATAFORMA

Los usuarios identificados que interactúan con la plataforma son los siguientes:

1. *Usuario Interno* (OTEC), es aquel usuario provisto por el OTEC, el cual utiliza la plataforma para realizar los procesos internos, como, por ejemplo: digitación de información, conformación de la orden de trabajo, creación de pruebas, y carga material de apoyo, entre otros. Este usuario es el principal agente que interactúa con la plataforma desarrollada.
2. *Usuario Alumno*, es aquel usuario que será capacitado por el OTEC, a pesar de que es un usuario secundario. Éste interactúa con la plataforma en diversos procesos, como por ejemplo en la obtención de sus credenciales, acceso al diploma, acceso al material de apoyo de la capacitación, y obtención de la información de su proceso de capacitación en sí.
3. *Usuario Empresa* (cliente del OTEC), es aquel usuario que contrata el servicio de capacitación del OTEC para sus empleados. Este usuario tiene acceso a los servicios que le permiten obtener información de sus empleados a capacitar, acceso al estatus de la capacitación, y obtención de los diplomas y certificados de capacitación, entre otros.
4. *Usuario Relator*, es aquel usuario indicado por el OTEC, quién dictará la capacitación. Este usuario interactúa con la plataforma para obtener el listado de alumnos del curso de capacitación que le corresponde dictar, acceso a la toma de asistencia, obtención de material de apoyo para la capacitación, y acceso a material de protocolos, entre otros.

3.2 PRINCIPALES CASOS DE USO DE LA SOLUCIÓN

Para identificar los casos de uso de la solución, primeramente, se identificaron lo siguientes actores y sus objetivos (Tabla 1). Luego la Figura 2 muestra el diagrama general de casos de uso, y los actores involucrados.

Tabla 1. Actores y objetivos soportados por el sistema⁴

Actor	Objetivos
Usuario Interno:	Registrar Diplomas CU1
	Registrar Credenciales CU2
	Crear Pruebas CU3
	Revisar Pruebas CUS1
	Registrar Encuesta de Satisfacción CU4
	Revisar Encuestas de Satisfacción CUS2
	Registrar Alumnos CU5
	Revisar Rendición de Gastos CUS3
	Registrar Empresa CU6 Crear Certificados CU7
Usuario Alumno:	Responder Encuesta de Satisfacción CUS4
	Solicitar Diploma CUS5
	Solicitar Credenciales CUS6
	Solicitar Certificados CUS7
	Consultar Asistencia CUS8
Usuario Empresa:	Solicitar Capacitación CU8
	Solicitar Informe de Capacitación CUS9
Usuario Relator:	Consultar Material de Apoyo CUS10
	Registrar Asistencia CU9
	Registrar Rendición de Gastos CU10

⁴ Se indican en negrita los casos de uso primarios.

Tras identificar los casos de uso primarios y secundarios para cada uno de los actores del sistema, se prosigue con la especificación de dichos los casos de uso a alto nivel.

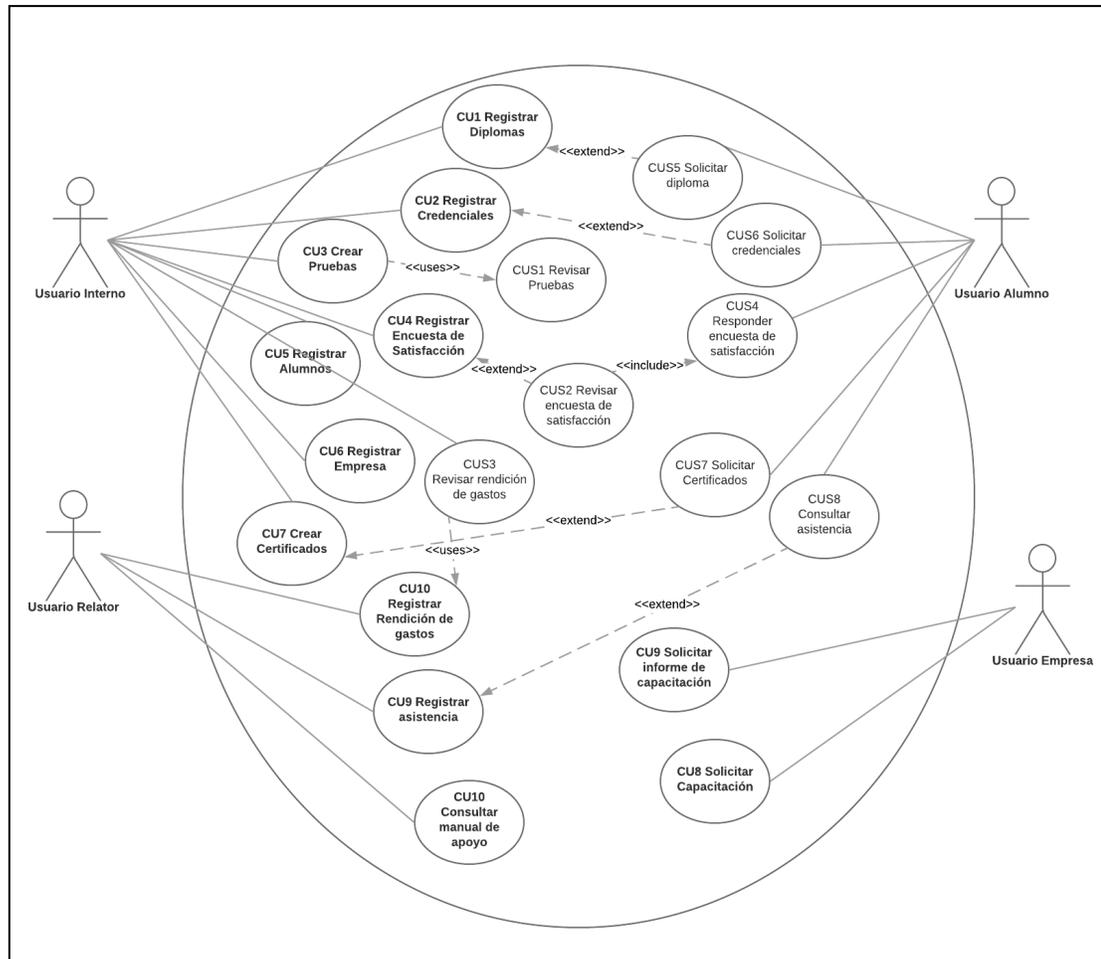


Figura 2. Diagrama casos de usos y sus actores.

3.3 CASO DE USO PRIMARIOS

Los casos de uso primarios identificados son los siguientes:

Caso de Uso	CU1: Registrar Diplomas
Actores	Usuario Interno (I)
Tipo	Primario
Descripción	Cuando un usuario alumno aprueba su capacitación, el usuario interno debe crear el diploma correspondiente y liberarlo en el sistema.

Caso de Uso	CU2: Registrar Credenciales
Actores	Usuario Interno (I)
Tipo	Primario
Descripción	El usuario interno deberá generar las credenciales en el sistema para cada usuario de tipo alumno que esté cursando una capacitación.
Caso de Uso	CU3: Crear Pruebas
Actores	Usuario Interno (I)
Tipo	Primario
Descripción	El usuario interno debe crear las pruebas correspondientes para cada curso de capacitación que ofrezca el OTEC.
Caso de Uso	CU4: Registrar Encuesta de Satisfacción
Actores	Usuario Interno(I)
Tipo	Primario
Descripción	El usuario interno debe crear las encuestas de satisfacción para cada curso de capacitación que ofrezca el OTEC.
Caso de Uso	CU5: Registrar Alumnos
Actores	Usuario Interno(I)
Tipo	Primario
Descripción	El usuario interno debe registrar a cada alumno que participará del curso de capacitación, según indique la orden de trabajo.
Caso de Uso	CU6: Registrar Empresa
Actores	Usuario Interno (I)
Tipo	Primario
Descripción	Cuando llega una nueva empresa solicitando los servicios de capacitación, el usuario interno debe registrar la empresa para poder generar una orden de trabajo.

Caso de Uso	CU7: Crear Certificados
Actores	Usuario Interno (I)
Tipo	Primario
Descripción	Una vez terminada la capacitación el usuario interno debe generar los certificados correspondientes, tanto para los alumnos como para la empresa.

Caso de Uso	CU8: Solicitar Capacitación
Actores	Usuario Empresa (I)
Tipo	Primario
Descripción	El usuario empresa debe poder solicitar los servicios de capacitación.

Caso de Uso	CU9: Registrar Asistencia
Actores	Usuario Relator (I)
Tipo	Primario
Descripción	El relator debe poder registrar la asistencia de cada alumno al inicio de la clase.

Caso de Uso	CU10: Registrar Rendición de Gastos
Actores	Usuario Relator (I)
Tipo	Primario
Descripción	El relator debe poder dejar un registro de los gastos realizados al dictar un curso de capacitación fuera de las dependencias del OTEC.

3.4 CASOS DE USO SECUNDARIOS

Los casos de uso secundarios identificados son los siguientes:

Caso de Uso	CUS1: Revisar Pruebas
Actores	Usuario Interno (I)
Tipo	Secundario

Descripción	Una vez rendidas las pruebas escritas por los alumnos, el usuario interno debe poder revisar y dejar registro de ello.
Caso de Uso	CUS2: Revisar Encuesta de satisfacción
Actores	Usuario Interno (I)
Tipo	Secundario
Descripción	El usuario debe poder revisar la encuesta de satisfacción para recopilar los datos y generar información del feedback dado.
Caso de Uso	CUS3: Revisar Rendición de gastos
Actores	Usuario Interno (I)
Tipo	Secundario
Descripción	El usuario debe poder validar los gastos registrados por los relatores.
Caso de Uso	CUS4: Responder Encuesta de Satisfacción
Actores	Usuario Alumno (I)
Tipo	Secundario
Descripción	El usuario debe poder responder la encuesta de satisfacción una vez terminado el curso de capacitación.
Caso de Uso	CUS5: Solicitar Diplomas
Actores	Usuario Alumno (I)
Tipo	Secundario
Descripción	El usuario debe poder solicitar su diploma en caso de haber aprobado el curso de capacitación.
Caso de Uso	CUS6: Solicitar Credenciales
Actores	Usuario Alumno (I)
Tipo	Secundario
Descripción	El usuario debe poder solicitar la creación de sus credenciales una vez terminado el proceso de capacitación

Caso de Uso	CUS7: Solicitar Certificados
Actores	Usuario Alumno (I)
Tipo	Secundario
Descripción	El usuario debe poder solicitar la generación de los certificados una vez terminado el proceso de capacitación.
Caso de Uso	CUS8:
Actores	Usuario Alumno (I)
Tipo	Secundario
Descripción	El usuario debe poder revisar su porcentaje de asistencia una vez que haya iniciado el proceso de capacitación.
Caso de Uso	CUS9: Solicitar Informe de Capacitación
Actores	Usuario Empresa (I)
Tipo	Secundario
Descripción	El usuario debe poder solicitar en cualquier momento, el informe de estado de la capacitación una vez que ésta comience.

3.5 PROCESOS A APOYAR

El principal proceso por apoyar en un OTEC es la ejecución de una capacitación. Este proceso es crítico ya que necesita un gran manejo de información, e involucra la gestión interna y externa del organismo. Por ello, en este proceso el sistema desarrollado contempla la participación de todos los usuarios declarados en la sección 0.

De este proceso se desprenden otros subprocesos. En específico, en cada interacción con el sistema se gatillan subprocesos con los que interactúan los otros tipos de usuarios. A continuación, se muestra el proceso realización de una capacitación.

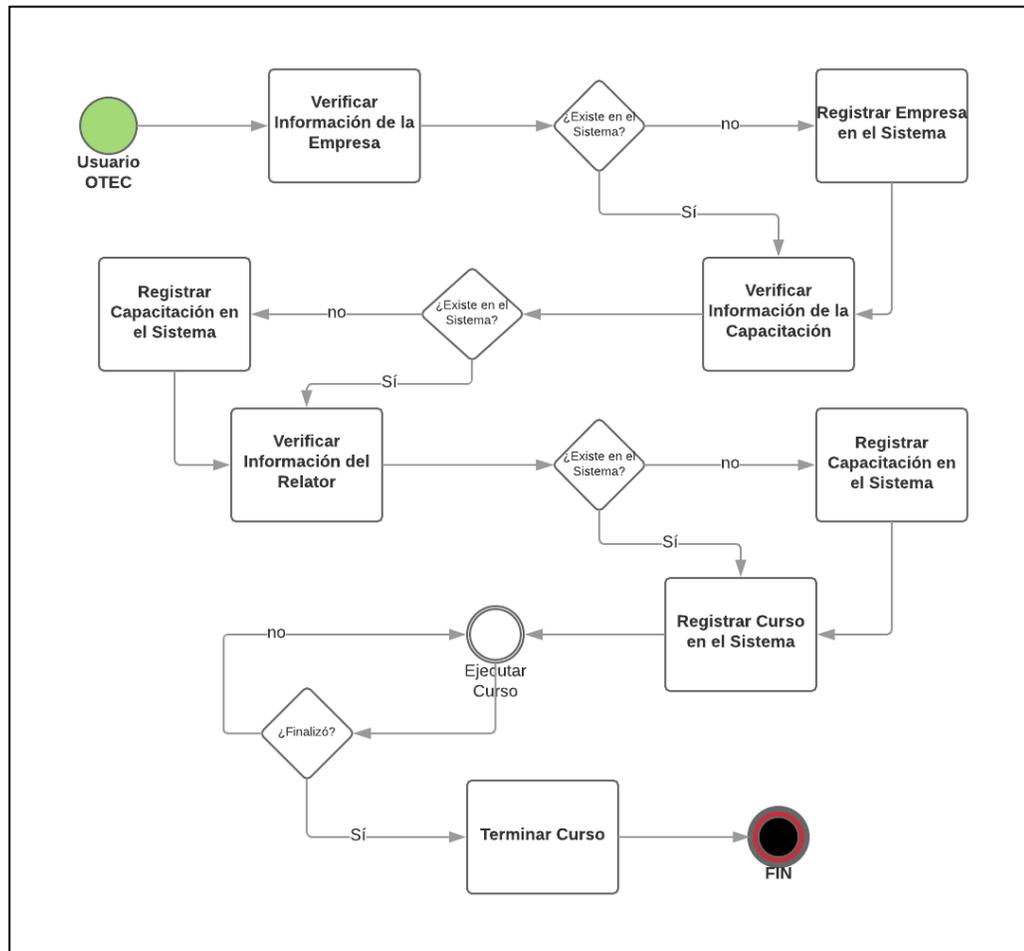


Figura 3. Proceso realización de una capacitación en el sistema web.

Proceso realización de una capacitación. En particular el proceso detallado de la Figura 3, corresponde al proceso de gestión interno de la ejecución de una capacitación por parte del OTEC. Allí el usuario administrador debe interactuar con los módulos de creación de empresa, creación de usuario relator y creación de curso en estado "pendiente". Teniendo esta información ingresada en el sistema el curso, ya puede ser "ejecutado" dicho proceso, pasando del estado pendiente al estado "en ejecución", y posteriormente, al finalizar el curso se pasa a un estado "terminado". De este proceso se desprenden los siguientes subprocesos:

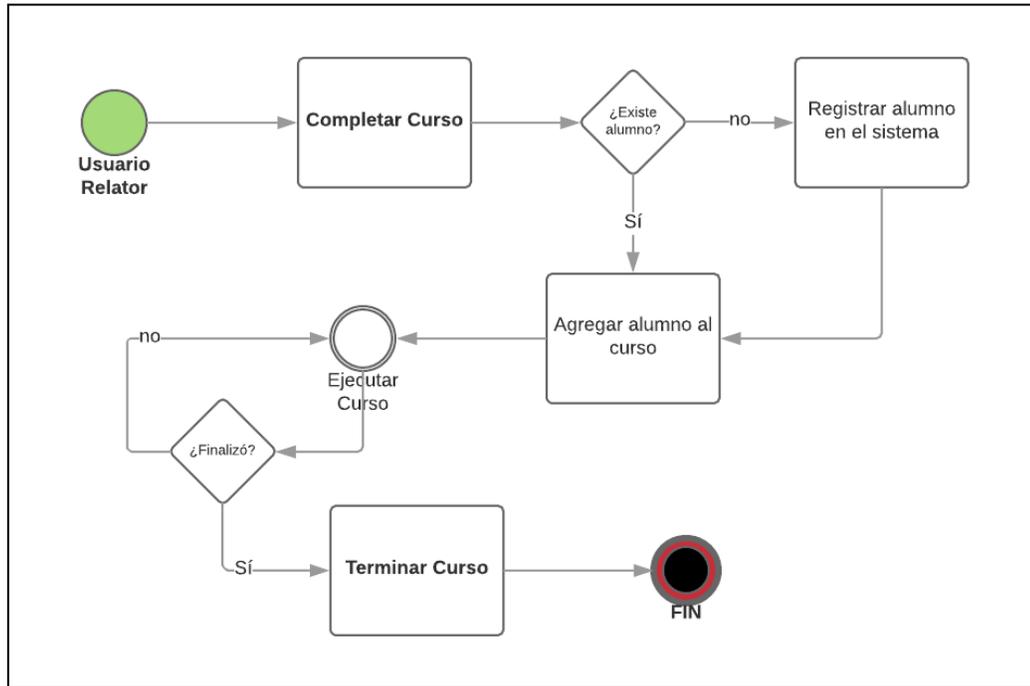


Figura 4. Proceso registro de alumnos para un curso.

Registro de alumnos para un curso. En este subproceso (Figura 4) el usuario relator interactúa con el sistema, con el fin de registrar a todos los usuarios estudiantes que participan en la capacitación solicitada por la empresa. Además, el usuario relator tiene la posibilidad de, una vez ejecutado el curso, cambiar su estado a "terminado"; esto con el fin de dar paso a la generación de credenciales.

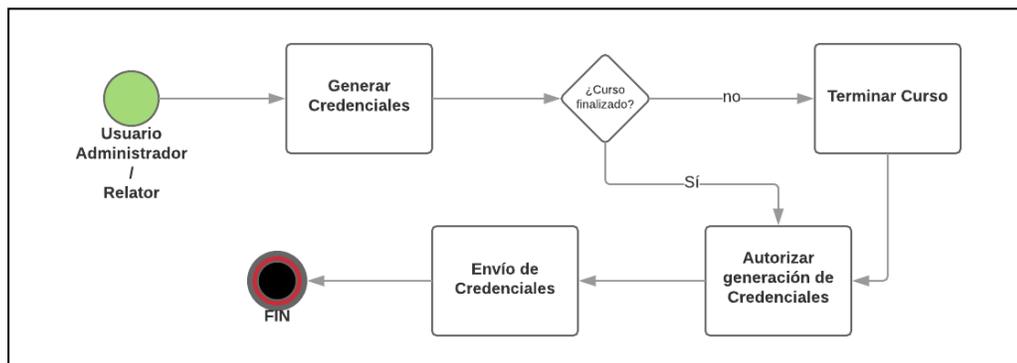


Figura 5. Proceso generación de credenciales.

Generación de credenciales. Tras la finalización de la ejecución de la capacitación, el usuario administrador y el usuario relator, tienen la facultad para iniciar el proceso de generación de credenciales. Este proceso es crítico debido a que cada credencial generada posee un identificador único.

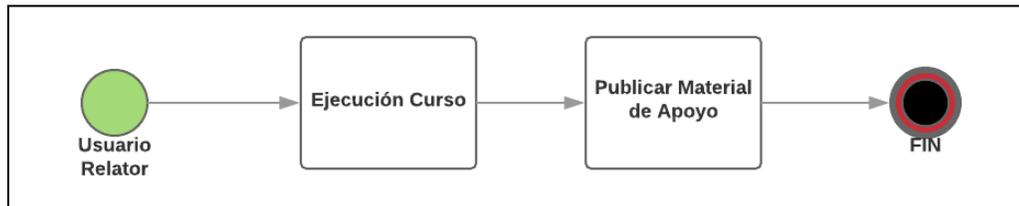


Figura 6. Proceso de publicación de material de apoyo.

Publicación de material de apoyo. Durante la ejecución de un curso de capacitación, el usuario relator puede generar y publicar en el sistema, el material de apoyo para los alumnos, como por ejemplo: guías de reforzamiento, y pruebas de ejemplo, entre otros.

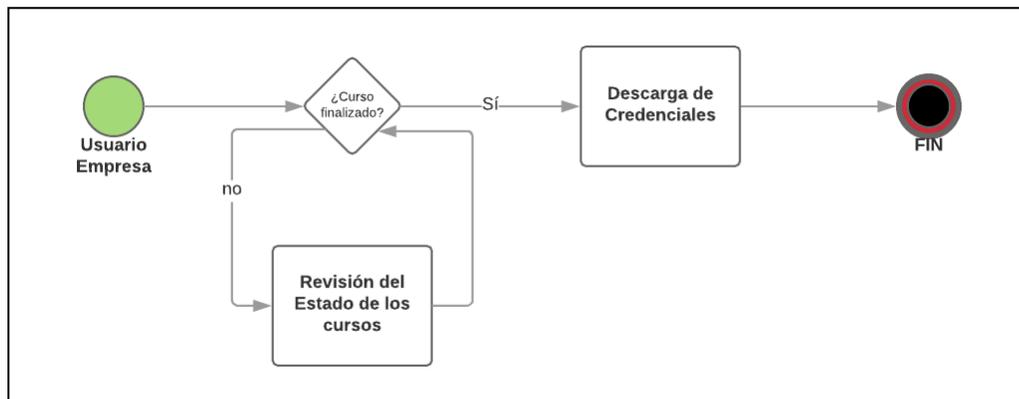


Figura 7. Proceso de solicitud de credenciales.

Solicitud de credenciales. Una vez dictado el curso, el usuario empresa puede descargar las credenciales de sus empleados (usuarios estudiantes); esto sólo si el curso ya se ejecutó y se encuentra en estado "terminado". En caso contrario, de este curso sólo se puede revisar su estado y la información de los empleados que figuran en el sistema.

Dados estos procesos, cabe destacar que todos ellos se desprenden de uno principal, o sea, del proceso de ejecución de una capacitación.

3.6 TECNOLOGÍAS ESCOGIDAS

Las principales tecnologías que se utilizaron en el desarrollo de la solución se separaron en tres segmentos: en framework de desarrollo para el eje central de la solución, framework para versionado y despliegue, y finalmente infraestructura en la nube para el despliegue productivo de la solución.

Dado que la solución consta particularmente de dos desarrollos independientes, el primero es el back-end y el segundo es el front-end. El primero fue desarrollado con Django, framework de código abierto que utiliza el lenguaje Python como soporte de base. Este framework es utilizado principalmente para el desarrollo de aplicaciones web, en conjunto a Django REST framework que es un kit de herramientas para construir API's Web. Este kit de herramientas es una librería que al ser añadida a cualquier proyecto Django, permite generar APIs en base a los modelos generados en dicho proyecto.

La elección de estas herramientas se basa principalmente en la flexibilidad que prestan para la generación de diferentes módulos del sistema, y para apoyar la escalabilidad de estos sin tener que alterar la arquitectura principal del proyecto construido. En ese sentido basta con añadir una nueva aplicación Django al proyecto principal, permitiendo a éste acceder a todos los modelos, funciones, vistas del proyecto principal, sin tener necesidad de preocuparse por generar adaptadores para ello.

Respecto al desarrollo del front-end se desarrolló el proyecto con el framework Angular⁵. Se eligió este framework, dado que uno de los principales desafíos del trabajo es generar un sistema que destaque en la usabilidad prestada a los usuarios. Es por ello que tras el análisis de los perfiles de usuarios que utilizan el sistema, se identificó que ellos están más relacionados a programas de escritorios, que con sistemas Web que trabajan online. Debido a que la solución que se desarrolló es un sistema online, se utilizó el desarrollo una aplicación de una sola página (en inglés, SPA: Single Page Application). Esta estrategia de despliegue de la aplicación le permite al usuario, vivir una experiencia más fluida y similar a una aplicación de escritorio, manteniendo las ventajas mencionadas en la sección 1.3 de una aplicación web. El paradigma SPA permite además cargar la mayoría de los recursos HTML, CSS y JavaScript al inicio de la interacción con el sistema. Luego el contenido de cada vista tiene una carga gradual según la interacción que va teniendo el usuario con el sistema.

En cuanto a las herramientas para el control de versiones, se utilizó GIT⁶, software de control de versiones utilizado para el mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Este control de versiones se trabajó utilizando Git-Flow⁷, que es un conjunto de extensiones de GIT que facilitan la gestión de ramas y flujos de trabajo. Esto facilitó la generación de releases y una fácil puesta en producción de estos, sin necesidad de eliminar los despliegues anteriores. En ese sentido, sólo basta con agregar las modificaciones sobre el código anterior utilizando GIT para ello.

⁵ <https://angular.io/>

⁶ <https://git-scm.com/>

⁷ <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

En el caso de la base de datos se utilizó PostgreSQL⁸. La elección del motor de base de datos es indiferente para la solución, ya que la arquitectura propuesta y la herramienta Django, permiten al desarrollador abstraerse de esta capa, pudiéndose cambiar el motor de base de datos sin necesidad de modificar o alterar otras capas del sistema.

Finalmente, como parte de la solución se utilizaron los servicios de AWS⁹ para la infraestructura donde se desplegó la solución. En este caso se utilizaron particularmente:

- *Amazon Elastic Compute Cloud* (Amazon EC2), servicio web que proporciona capacidad de computación en la nube. En este servicio se desplegaron los dos proyectos desarrollados de la solución.
- *Amazon Route 53*, servicio web DNS (Sistema de nombres de dominio) escalable y de alta disponibilidad. En este servicio se alojó el dominio .CL "otecpro.cl", dominio comprado en NIC Chile (entidad encargada de administrar el registro de nombres de dominio .CL).

El uso de infraestructura en Amazon permite la escalabilidad del sistema, ya que en la medida que la aplicación necesite recursos adicionales para su funcionamiento, basta con aumentar la configuración de esta plataforma para permitir de forma transparente un aumento de la capacidad de atención de un OTEC.

3.7 DISEÑO DE LA ARQUITECTURA

⁸ <https://www.postgresql.org/>

⁹ <https://aws.amazon.com/es/>

Con respecto a las tecnologías a usar en el desarrollo de la plataforma web, se escogió el framework Django para el back-end, ya que el proponente posee conocimientos previos de la herramienta en cuestión, lo cual facilita la curva de aprendizaje de dicha tecnología. Considerando la elección de la herramienta principal de desarrollo, el diseño de la arquitectura (Figura 8) se desglosa de la siguiente manera:

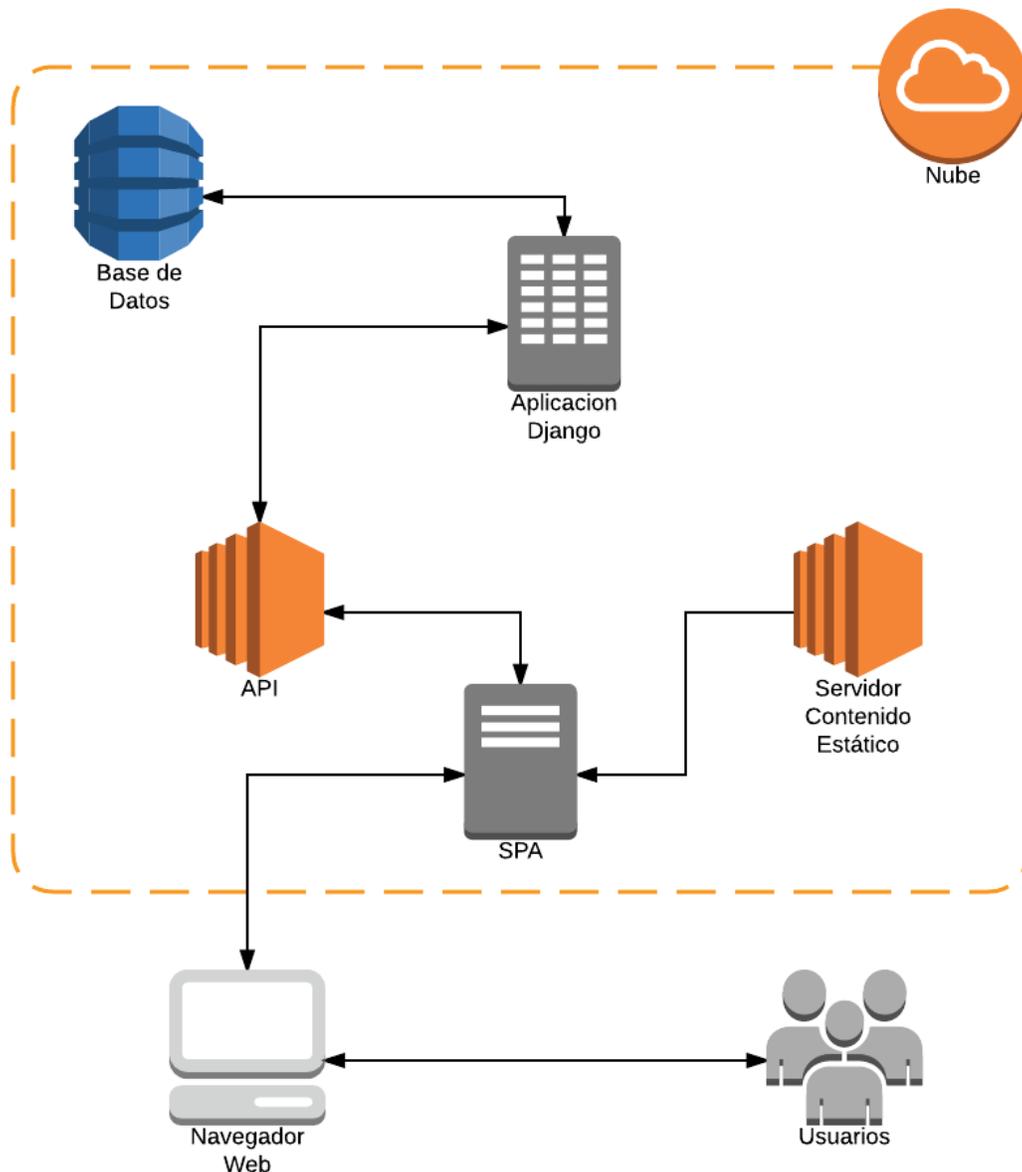


Figura 8. Arquitectura de la solución

1. *Back-end*. Como se mencionó antes, para el back-end o capa lógica se utilizará el framework de código abierto Django, que usa el lenguaje Python como base. Este framework es muy utilizado para el desarrollo de aplicaciones Web.
2. *Front-end*. Para el front-end o capa de usuario se utilizará el framework Angular, bajo el lenguaje TypeScript. Este framework es frecuentemente usado en el desarrollo de aplicaciones de una sola página (SPA). La arquitectura del front-end les permite a los usuarios vivir una experiencia similar a la de una aplicación de escritorio.
3. *API REST*. Para el desarrollo de la capa intermedia entre el front-end y el back-end propuesto, se utilizó una API (Application Programming Interface), que involucra un conjunto de reglas que permiten la comunicación con los componentes (o servicios) incluidos en ésta. La API fue implementada utilizando la librería Django REST Framework, que es una aplicación Django que permite construir proyectos bajo la arquitectura REST.
4. *Capa de Datos*. Como se mencionó antes, el motor de base de datos utilizado es PostgreSQL. Sin embargo, la elección del motor de datos es indiferente para la plataforma considerada en esta solución.

Finalmente, para el desarrollo de los módulos propuestos se utilizó la arquitectura provista por Django, la cual permite dentro de un proyecto principal, generar aplicaciones como módulos de éste. Estos proyectos pueden compartir los modelos y la lógica creada en cada una de las aplicaciones. Es decir, cada módulo será una aplicación Django dentro del proyecto principal, que corresponderá a la plataforma web para la gestión del OTEC.

Dentro de la identificación de los usuarios de la plataforma, se identificaron las prioridades de los diferentes módulos propuestos para el desarrollo. Esto permite generar una implementación y validación incremental, que brinde un apoyo razonable a las operaciones de un OTEC.

3.8 MODELO DE DATOS

El modelo de datos de la solución se denota por el siguiente Diagrama Entidad-Relación.

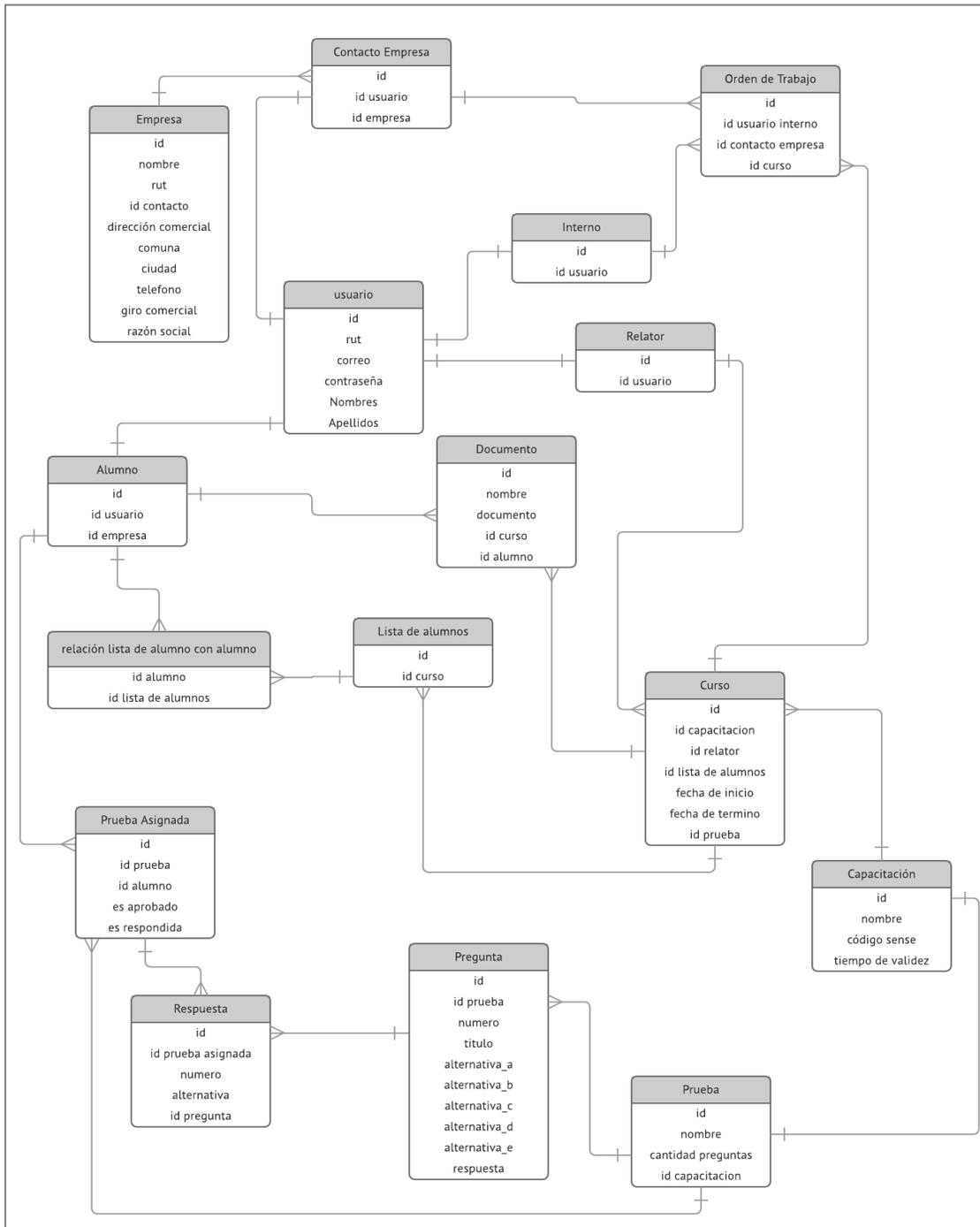


Figura 9. Diagrama Entidad Relación de la Base de Datos del sistema.

A continuación, se detallan las principales tablas del diagrama mostrado en la Figura 9, y sus atributos más relevantes:

1. *Empresa*: La empresa es el ente que solicita los servicios de capacitación del OTEC.
2. *Usuario*: Corresponde a la entidad usuario que podrá interactuar con el sistema; éste se desglosa en los siguiente otros cuatro perfiles: contacto empresa, relator, alumno y usuario interno. El Contacto Empresa representa a la empresa que solicita los servicios de capacitación del OTEC. El Relator corresponde al usuario provisto por el OTEC que dictará la capacitación. El Alumno corresponde al usuario que recibirá la capacitación, y que pertenece a una empresa. Finalmente, el Usuario Interno corresponde al usuario empleado del OTEC, que gestionará toda la información dentro del sistema.
3. *Orden de trabajo*: Corresponde al documento en el cual se deja registro del servicio solicitado por la empresa al OTEC. En él se registra la empresa, el curso de capacitación a ejecutar, y el usuario interno gestionará la información, entre otros.
4. *Curso*: Corresponde al curso de capacitación que se dictará según se indique en la orden de trabajo. El curso posee un relator, una fecha de inicio y término, y una lista de alumnos.
5. *Capacitación*: Esta entidad contiene el código de registro SENCE, y posee una prueba asignada para evaluar a los alumnos posterior a su capacitación.
6. *Documento*: Representan los diplomas y certificados según le corresponda a cada alumno, en función de si ha aprobado o reprobado su evaluación.
7. *Prueba*: Corresponde a la evaluación de una capacitación. Cada alumno perteneciente a un curso de debe rendir dicha prueba y aprobarla para recibir su certificación de capacitación.

3.9 COMPONENTES DE NEGOCIO

A continuación se muestran, y luego se describen, los principales componentes de negocio involucrados en la solución desarrollada (Fig. 10).

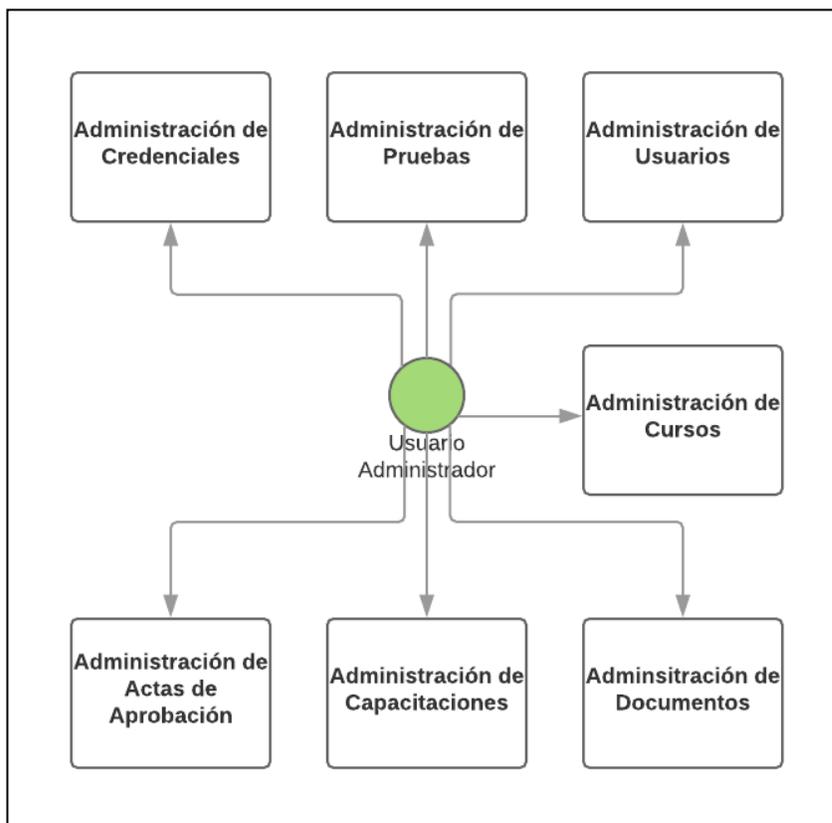


Figura 10. Módulo Administración y sus submódulos.

El eje central de la solución es el módulo de administración (Figura 10) para la gestión del OTEC. En él el usuario OTEC, puede gestionar los usuarios que utilizarán la plataforma; entre ellos se encuentra los usuarios Relator, Empresa y Estudiante, pudiendo crear y modificar datos de estos.

El componente (o submódulo) de administración de cursos, permite crear y editar un curso, pudiendo agregar estudiantes a éste, asignarles una prueba, una capacitación y un relator. Tras completar estos datos el curso, éste se encuentra en estado pendiente hasta que sea ejecutado por el usuario relator, y luego finalizado para continuar con la generación de credenciales.

El componente administración de documentos, permite subir archivos en distintos formatos para apoyar la labor de capacitación, ya sean lecturas complementarias, pruebas de ejemplo, guías de apoyo, entre otros. Permitiendo se descargadas por todos los usuarios del sistema.

El componente administración de capacitaciones, permite crear y editar capacitaciones con los datos asignados por el SENCE, como es: código SENCE, fecha de acreditación y fecha termino de vigencia.

El componente de administración de actas de aprobación, permite subir un archivo digitalizado de un acta física en la que se enlistan los estudiantes que hayan aprobado el curso. Este documento es brindado al usuario empresa, el cual necesita tener un acceso rápido a este documento que acredita que los estudiantes que aquí aparecen están capacitados, en caso de que sean fiscalizados y necesiten evidencia para comprobar que los estudiantes están en vigencia con las capacitaciones necesarias para cumplir sus labores.

El componente de administración de credenciales, permite subir un archivo digitalizado de la tarjeta o credencial que indica en forma clara que capacitación está vigente para un estudiante en particular, al igual que el acta de aprobación, este es entregado al usuario empresa con el mismo objetivo.

El componente administración de pruebas, permite crear y editar pruebas digitales, mediante preguntas de alternativas múltiples, con el fin de realizar una prueba inicial que le permita al estudiante comprobar sus conocimientos previos a la prueba presencial requerida para aprobar el curso.

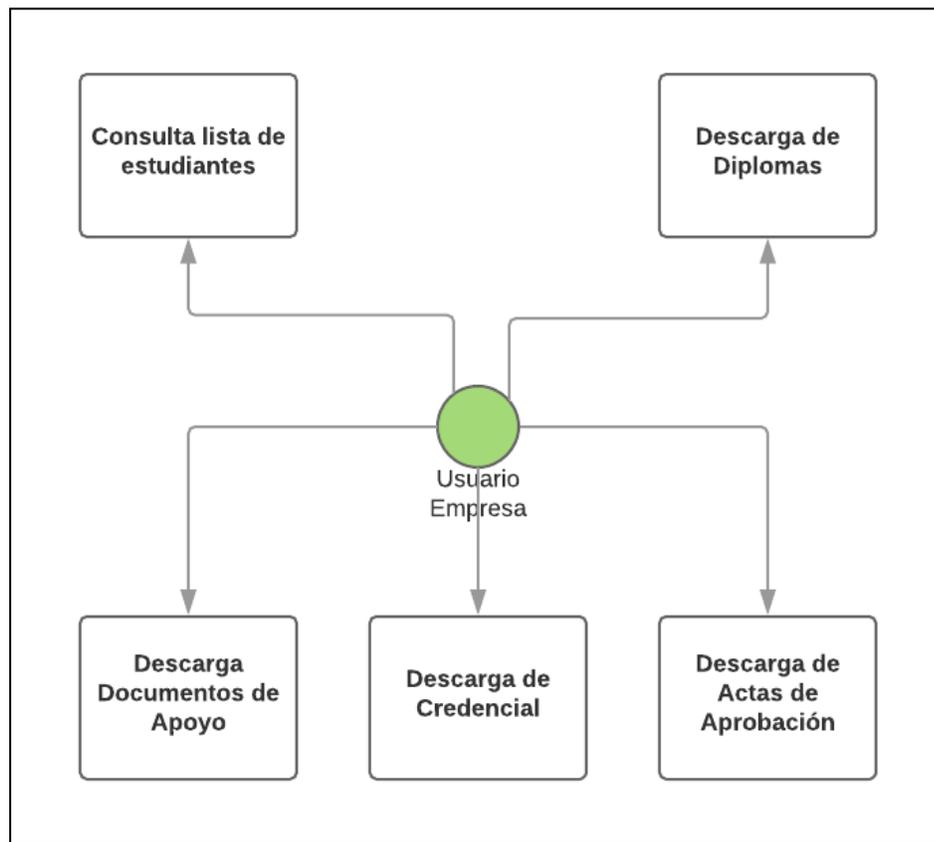


Figura 11. Módulo Empresa y sus submódulos.

Respecto a los módulos con los que interactúa el usuario empresa (Figura 11) encontramos los siguientes componentes (o submódulos):

Componente de descarga de diploma, que permite a la empresa acceder a los diplomas de todos los estudiantes pertenecientes a la empresa, pudiendo descargar los documentos que validan la aprobación de los estudiantes según el curso rendido.

Componente de descarga de credenciales, permite a la empresa acceder a las credenciales de todos los estudiantes pertenecientes a la empresa, pudiendo descargar dicha credencial en su versión digitalizada. Allí se indica de forma clara qué capacitación está vigente para un estudiante en particular.

Componente de descarga de documentos de apoyo, permite a la empresa acceder a los documentos de apoyo de un curso en particular. Misma información a la que accede un usuario estudiante.

Componente de consulta de estudiante, permite a la empresa revisar el listado de estudiantes que están (o participaron) de alguna capacitación con el OTEC, ya sea, revisando estudiante por estudiante o curso por curso.

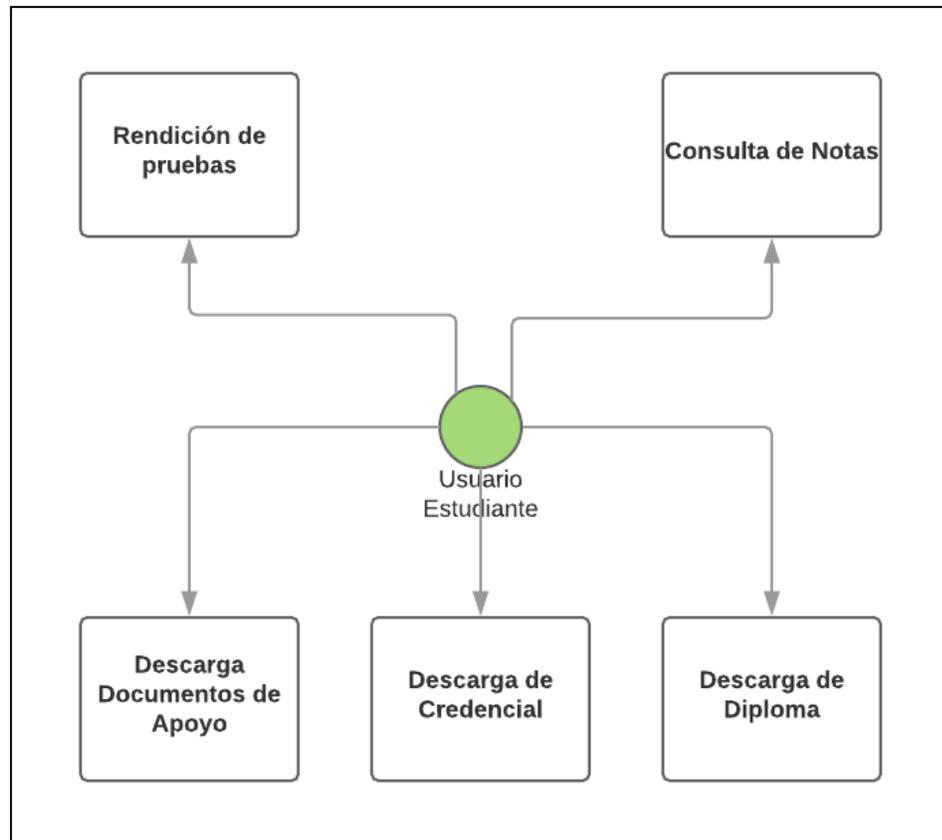


Figura 12. Módulo Estudiante y sus submódulos.

Respecto a los submódulos con los que interactúa el usuario estudiante (Figura 12) encontramos a los siguientes:

Componente de rendición de pruebas, el cual permite revisar las pruebas asignadas al estudiante según su participación en los diversos cursos. Aquí el estudiante responde una prueba online de selección múltiple.

Componente de consulta de notas, permite al estudiante hacer seguimiento de las notas de las pruebas online y de la prueba presencial. Estas notas son cargadas al sistema por el OTEC.

Componente de descarga de diploma, permite al estudiante, según haya aprobado el curso, acceder a la descarga de su diploma; documento que acredita su aprobación al curso rendido.

Componente de descarga documentos de apoyo, permite al estudiante acceder a una serie de documentos de apoyo a la capacitación que está rindiendo, ya sean, lecturas complementarias, pruebas de ejemplo, material de apoyo, etc.

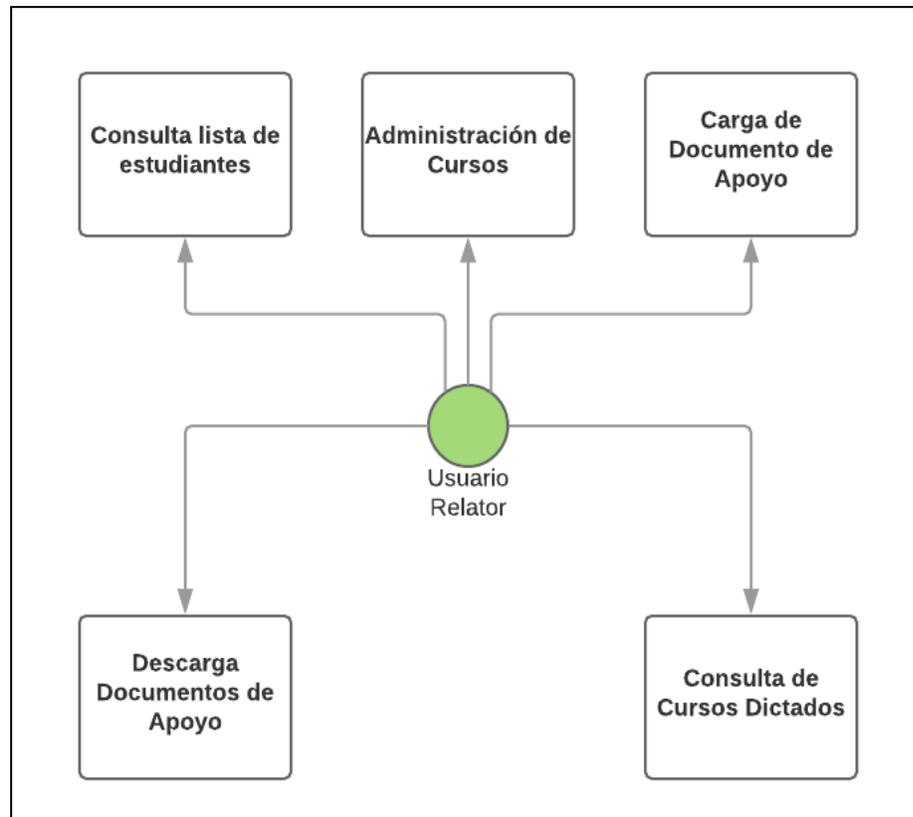


Figura 13. Módulo Relator y sus submódulos.

Respecto a los submódulos con los que interactúa el usuario relator (Figura 13) encontramos los siguientes:

Componente de carga de documentos de apoyo, que permite al relator subir los archivos de apoyo a un curso, los cuales son descargados por los usuarios empresa y estudiantes.

Componente de consulta de cursos dictados, permite al relator hacer seguimiento en los cursos a los cuales está asignado como relator, pudiendo ver el estado del curso y la información de éste.

Componente de descarga de documentos de apoyo, al igual que el usuario empresa y estudiante, el relator puede descargar los documentos de apoyo subidos por otros usuarios al sistema.

Componente de consulta de listado de estudiantes, permite al relator revisar, y buscar información de cada estudiante perteneciente al curso que dictará. Esto se realiza para corroborar la información en línea, en caso de ser necesario.

Componente de administración de cursos, a diferencia de los permisos que posee un usuario administrador, en este caso al relator sólo se le permite consultar la información del curso y cambiar el estado de éste, ya sea, cambiarlo el estado del mismo (de "pendiente" a "en ejecución", o bien a "terminado"), cuando éste ya ha sido dictado.

3.10 ESTRUCTURA DEL BACK-END

El back-end de la solución está constituido como un proyecto Django, el cual posee una aplicación que contiene toda la abstracción del modelo de datos; que consiste en el mapeo del modelo entidad-relación, a una estructura de clases. En el back-end también encontramos los serializers, que son clases (y sus métodos) permiten transformar datos complejos, como por ejemplo, la transformación de los resultados de un conjunto de consultas e instancias de modelos, a los tipos de datos nativos de Python, los cuales posteriormente se representan en archivos en formato JSON. Además de estas clases, también se encuentran la serialización que hacen el proceso inverso, una vez que se haya validado la información.

Adicionalmente, el back-end de la solución se encarga de almacenar y gestionar los documentos del OTEC, como lo son los diplomas, imágenes, y documentos de apoyo, entre otros. Los servicios de interacción con estos documentos también se encuentra disponible a través de la API.

Para que sea posible interactuar con las entidades de la solución a través del front-end, es necesario controlar quienes interactúan con la API, ya que ésta es pública. Para ello se utiliza un mecanismo de autenticación basado en un *Token*, el cual es entregado al usuario cada vez que éste inicia sesión utilizando sus credenciales (correo electrónico y contraseña). Este *Token*, es utilizado en toda solicitud HTTP hacia la API, lo que permite identificar al usuario que

realiza la solicitud, y junto con ello, permite controlar los permisos y gestionar qué información puede leer, crear, editar y/o eliminar dicho usuario.

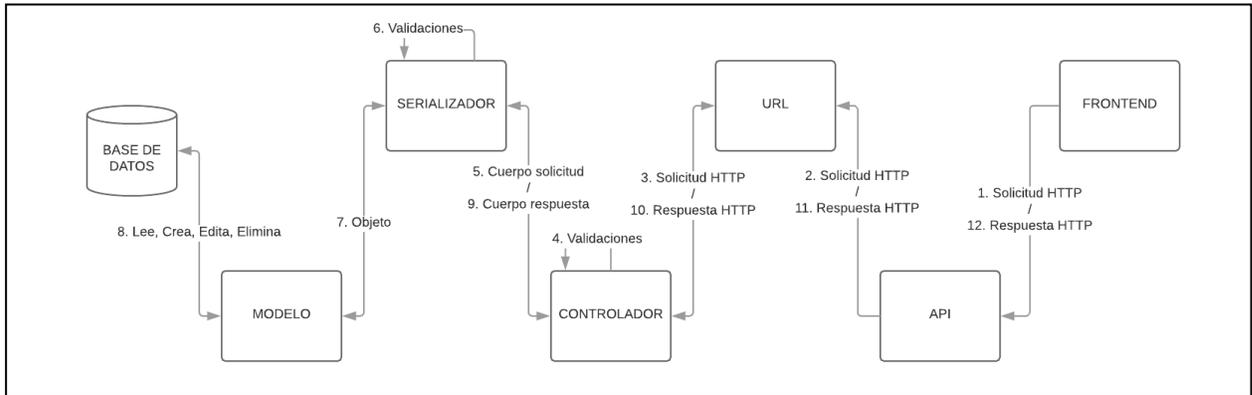


Figura 14. Diagrama Funcionamiento Back-end.

El *Token* debe ir incluido en la cabecera *Authorization* de la solicitud HTTP hacia la API (punto 1 en la Figura 14). Esta solicitud es creada en el front-end, y luego llega a la colección de URLs (punto 2). Aquí es recibida por un controlador encargado de procesar la url desde la cual se está generando la solicitud. En este controlador se realizan diversas validaciones (punto 4) como es verificar que el usuario posea los permisos correctos para acceder a la url, y realizar una de las 4 opciones que permite el sistema (leer, crear, editar y/o eliminar).

Una vez que se cumplen las validaciones, el contenido o cuerpo de la solicitud (punto 5) es serializado y validado según las reglas lógicas generadas para la url. Cumpliendo esto, el cuerpo de la solicitud es transformado en un *Objeto* (punto 7). Este objeto es generado según las reglas del modelo al que pertenece, este mapeo permite que el objeto se transforme en un registro de un modelo. Luego, una vez generado el modelo, éste es llevado a la Base de Datos (punto 8); en el caso de un acceso de lectura, el objeto es complementado con los datos solicitados. En el caso de una edición, el registro es actualizado con los datos del objeto. Finalmente, en el caso de una operación de creación, se crea un registro con los datos del objeto, y para una eliminación se utiliza la referencia al objeto en el registro que posee la base de datos (se elimina la referencia).

Una vez realizada la acción solicitada, comienza la construcción de la respuesta hacia el front-end. Con ello el objeto vuelve al serializador (punto 7), para validar si la creación, lectura, edición o eliminación fue exitosa. Este objeto, ahora serializado, es entregado al controlador (punto 8), ya conformado como el cuerpo de la respuesta a la solicitud. En el controlador se realizan las validaciones para verificar que se cumpla lo necesario para responder a la solicitud; esto es, vuelve por la misma url (punto 10). La respuesta ya contiene el código de respuesta, 200 para lectura, 201, creación, 202 edición y 203 eliminación. Luego la respuesta viaja por la API (punto 11) y finalmente arriba al front-end (punto 12), donde llega la respuesta a la solicitud. Este flujo describe cómo interactúa en general el proyecto front-end con el proyecto back-end, los cuales conforman en conjunto la solución.

3.11 API IMPLEMENTADA

La API de la solución contiene todos los servicios para la comunicación entre el front-end y el back-end. Esta interfaz de comunicación está construida en base a la librería Django REST Framework, y permite construir una aplicación bajo la arquitectura REST. Esto último se cumple buscando hacer que la API implementada sea escalable para permitir la modificación de los módulos desarrollados, y la adición de nuevos módulos en una etapa posterior a este proyecto. Las estructuras básicas de las url expuestas en la API de la solución son las siguientes:

1. Verbo_Http+https://{url_base}/api/v1/{entidades}/.
2. Verbo_Http+https://{url_base}/api/v1/{entidad}/{identificador_entidad}.
3. Verbo_Http+https://{url_base}/api/v1/{entidad}/{id_entidad}/{entidad_hija}/{id_hija }/.

En la composición de las URLs vemos al menos 3 componentes:

- a. *url_base*. Esta variable corresponde al dominio donde se encuentra alojado el proyecto Django; en este caso es www.otecpro.cl, por lo que el comienzo de la url es <https://www.otecpro.cl>.
- b. *Entidad o entidad_padre*. Corresponde a una entidad o actor principal del modelo de datos. En caso de que estemos en el módulo de usuarios estudiantes del sistema, esta variable sería `usuarios_estudiantes`, por lo

que la url ahora es https://www.otecpro.cl/usuarios_estudiantes. Con esta url, utilizando el método HTTP Get, obtendríamos el listado de usuarios estudiantes que pertenecen al sistema.

- c. *Id_entidad*. Corresponde al identificador de la entidad o actor principal del modelo de datos, en el caso de que deseemos modificar un usuario estudiante. Particularmente, en el caso del identificador numérico 7, la url ahora es https://www.otecpro.cl/usuario_estudiante/7/. Las modificaciones se realizan utilizando el método HTTP Update, y entregando en el body de la petición, los datos del usuario ya actualizados. Estos son entregados y validados por los controladores de la url, como se detalló en la lógica del back-end (Figura 14).
- d. *Entidad_hija*. Corresponde a una entidad que depende de otra según el modelo de datos. En el caso de que necesitemos las pruebas que pertenecen a un curso, la url sería <https://www.otecpro.cl/curso/5/pruebas/>. Utilizando el método HTTP Get obtendremos la lista de pruebas que pertenecen al curso, cuyo identificador en la base de datos es 7.

Cabe destacar que toda url está protegida por el token obtenido al iniciar sesión en el sistema. Con este token, en cada petición que pasa por la API, se verifica que el usuario que generó la petición tenga acceso a los datos y/o funciones que está solicitando. Además, esto permite agregar datos de validación que no están incluidos en las interacciones con la base de datos.

Por otro lado, la API permite al front-end utilizar las funciones básicas leer, crear, modificar y eliminar datos de la base de datos, previa validación del back-end. Además, existen funciones adicionales que se diferencian de las acciones básicas, siguiendo la estructura de conformación de url, la que permite de forma explícita modificar entidades de la base de datos, sin tener que agregar información adicional en la petición HTTP.

Finalmente, la API cumple con el hecho de que: cada mensaje HTTP contiene toda la información necesaria para entender la solicitud; la API posee operaciones que se encuentran bien definidas, estas son: leer, crear, modificar y eliminar; la estructura para identificar los recursos es únicamente

direccionable a través de su url. Con esto se cumple que la API desarrollada está bajo la arquitectura REST.

4 INTERFACES DEL SISTEMA

Dado que en el sistema web existen cuatro perfiles de usuarios distintos, es necesario detallar el mapa de navegación para estos usuarios de forma independiente. La Figura 15 muestra el mapa de navegación para el usuario administrador.

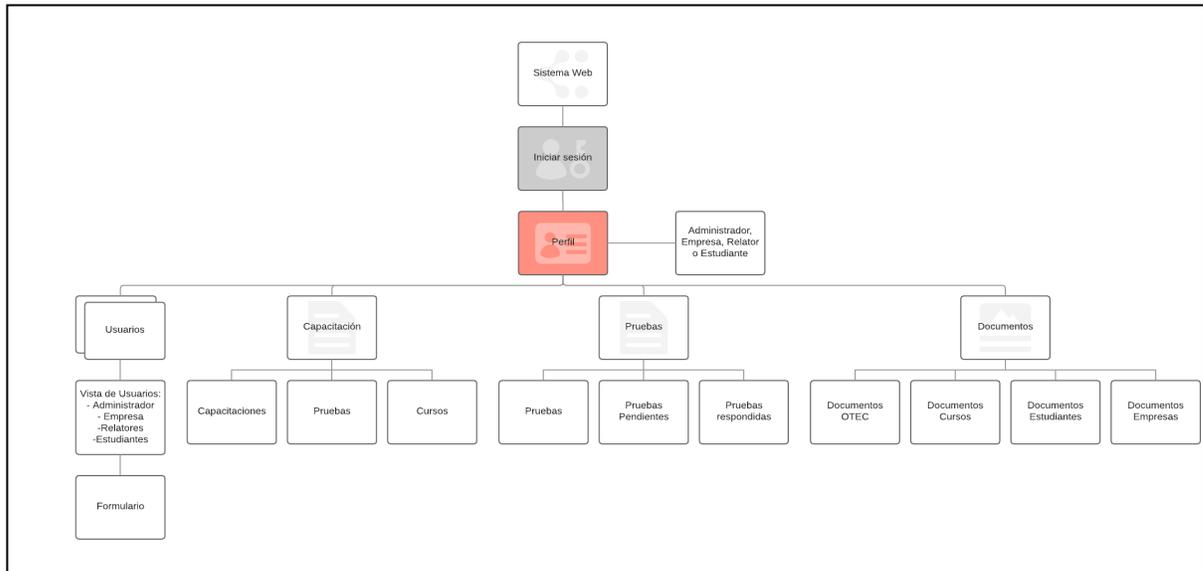


Figura 15. Mapa de navegación de la aplicación web para el perfil Administrador.

Las Figura 16 a 18 muestran el mapa de navegación para los tipos de usuario relator, estudiante y empresa respectivamente.

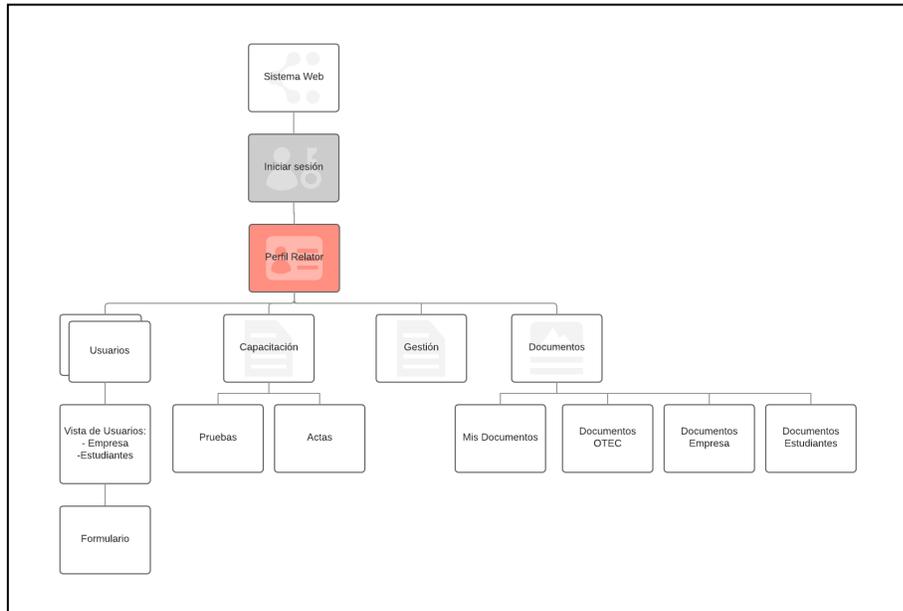


Figura 16. Mapa de navegación de la aplicación web para el perfil Relator.

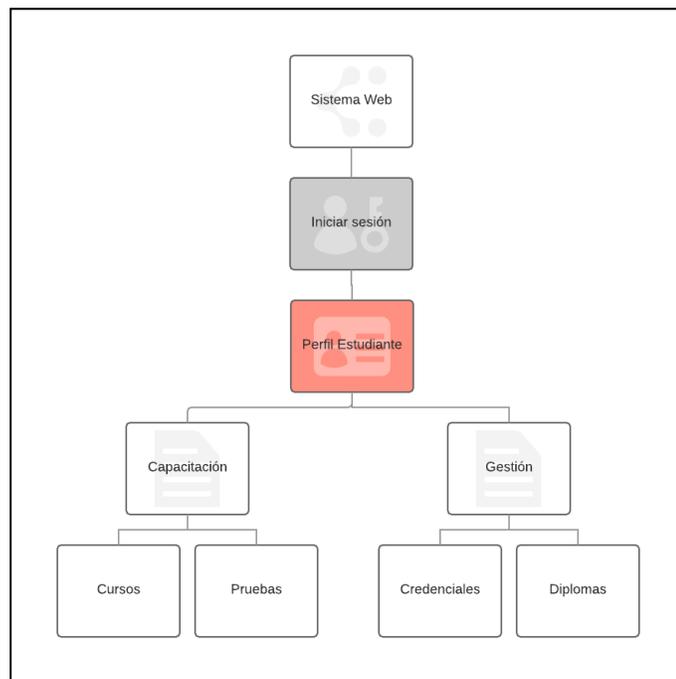


Figura 17. Mapa de navegación de la aplicación web para el perfil Estudiante.

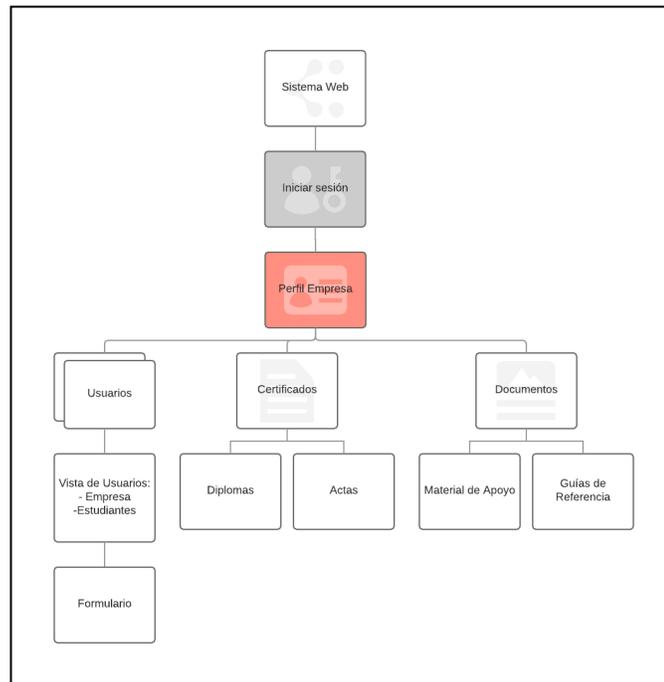


Figura 18. Mapa de navegación de la aplicación web para el perfil Empresa.

Por otro lado, las vistas del sistema están construidas en torno al eje principal de cada interacción que tienen los perfiles detallados anteriormente. Estos acceden a distintas vistas y módulos del sistema según el tipo de usuario. Por ende, a continuación, se ilustra las interacciones que posee cada tipo de usuario, con las vistas asociadas a la descripción de los flujos donde éste participa.

4.1 INTERFACES DEL USUARIO ADMINISTRADOR

Las principales interfaces con las que el usuario administrador interactúa tienen que ver con la ejecución del flujo principal del negocio; es decir, con ejecutar una capacitación. En primera instancia el usuario debe iniciar sesión en el sistema con sus credenciales. Luego debe continuar, con el registro de la empresa en el sistema; para ello debe dirigir, en el menú lateral izquierdo, la opción "Agentes" (Figura 19).

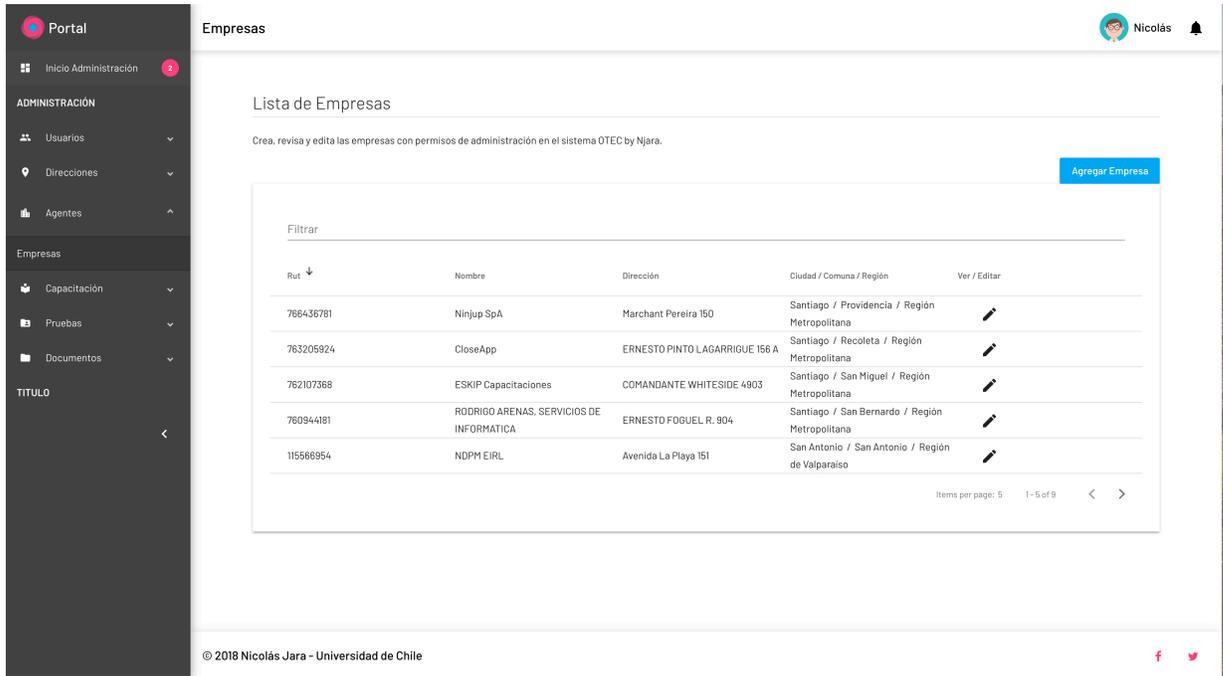


Figura 19. Vista lista de empresas.

Para agregar la empresa al sistema, el usuario debe completar el formulario de creación de empresa (Figura 20).

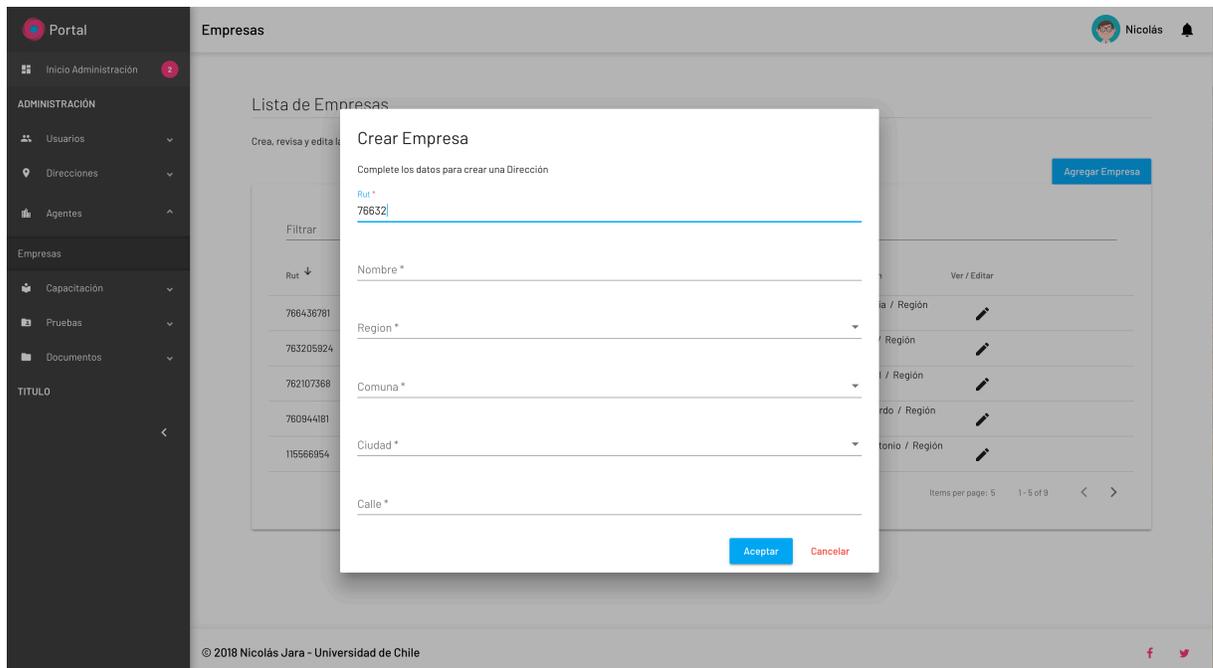


Figura 20. Formulario de registro de empresa.

Tras la creación de la empresa, se debe continuar con el registro de la capacitación. Para ello el usuario se dirige a la sección "Capacitaciones" y completa el formulario de creación de la capacitación. Este formulario posee el mismo diseño de la Figura 20, pero con distintos requisitos.

De la misma forma se realiza el registro del usuario Relator y de los Estudiantes. Tras ello, se debe registrar el curso en el sistema, para ello el administrador debe dirigirse a la sección "Cursos" (Figura 21). Allí debe completar el formulario de registro de curso; formulario que posee el mismo diseño de la Figura 20, pero también con distintos requisitos.

Portal

Cursos

Nicolás

Inicio Administración

ADMINISTRACIÓN

Usuarios

Direcciones

Agentes

Capacitación

Capacitaciones

Pruebas

Cursos

Libro de Clases

Facturas

Pruebas

Documentos

TÍTULO

Lista de Cursos

Crea, revisa y edita los Cursos con permisos de administración en el sistema OTEC by Njara.

Agregar Curso

Filtrar

Nombre	Año	Capacitación	Empresa	Relator	Estudiantes	Pruebas	Ver / Editar	Publicar
Curso 1	2018	CERO DAÑO, INDUCCION	Coca Cola	Johaziel	2	1		Finalizado ✓
Curso 2	2018	ADMINISTRACION DE EMERGENCIAS	Sodimac	Alejandro	1	1		Finalizado ✓
Curso 3	2018	ADMINISTRACION DE EMERGENCIAS	Otec	Alejandro	5	1		Finalizado ✓
Curso 4	2018	CERO DAÑO, INDUCCION	Sodimac	Johaziel	1	1		Finalizado ✓
Curso 5	2018	CERO DAÑO, INDUCCION	Sodimac	Alejandro	1	1		Pendiente ▶

Items per page: 5 1 - 5 of 6

© 2018 Nicolás Jara - Universidad de Chile

Figura 21. Vista sección cursos.

Finalmente, tras el registro de todos los participantes de un curso, se procede a agregar a los estudiantes al curso, mediante el formulario mostrado en la (Figura 22), para luego ejecutar el curso.

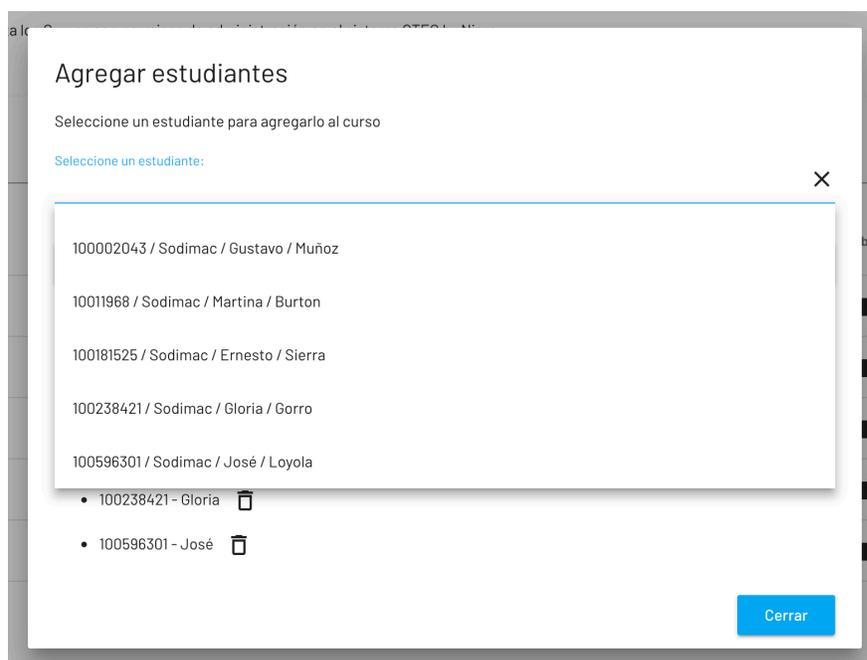


Figura 22. Formulario para agregar alumnos a un curso.

En este formulario, se agregan todos los alumnos que son participantes del curso, esto permite al usuario Relator, poner al curso en estado de ejecución, el curso sólo puede ejecutarse si posee al menos un alumno.

Finalmente, el curso una vez dictado, se cambia su estado a terminado, finalizando con ello el proceso de capacitación dentro del sistema web.

4.2 INTERFACES DEL USUARIO RELATOR

Las principales interfaces con las que el usuario relator interactúa en el sistema, tienen que ver con continuar con una ejecución de una capacitación, donde éste necesita crear una prueba para dicha capacitación. En primera instancia el usuario debe iniciar sesión en el sistema con sus credenciales, y luego continuar, registrando una prueba el sistema. Para ello debe dirigirse al menú lateral izquierdo donde se encuentra la sección "Pruebas", en donde visualizará una lista de pruebas con el mismo diseño gráfico de la Figura 19. Sin embargo, este formulario presenta una componente única en su diseño, como se indica en la Figura 23, debido a que las pruebas son de selección múltiple. Debido a eso, el formulario posee un wizard para cada pregunta, el cual solicita el título, las 5 alternativas de respuesta, y la alternativa correcta.

U
a la

Crear Prueba

Complete los datos para crear una Prueba

- Pregunta Nº 1
- Pregunta Nº 2
- Pregunta Nº 3
- 4** Pregunta Nº 4

Titulo *
¿Qué precaución se debe tener al trabajar en altura?

Alternativa A *
Revisar el casco

Alternativa B *
Revisar la línea de vida

Alternativa C *
Revisar el uniforme y zapatos.

Cerrar

Figura 23. Formulario para crear una prueba y sus preguntas.

4.3 INTERFACES DEL USUARIO ESTUDIANTE

Las principales interfaces con las que el usuario estudiante interactúa con el sistema es al revisar los cursos donde está participando. En primera instancia el usuario debe iniciar sesión en el sistema con sus credenciales, y continuar revisando y verificando los cursos donde éste está participando. Para ello el usuario debe dirigirse al menú lateral izquierdo, donde se encuentra la sección "Cursos". Allí se visualizará una lista de los cursos donde el estudiante participa (Figura 24).

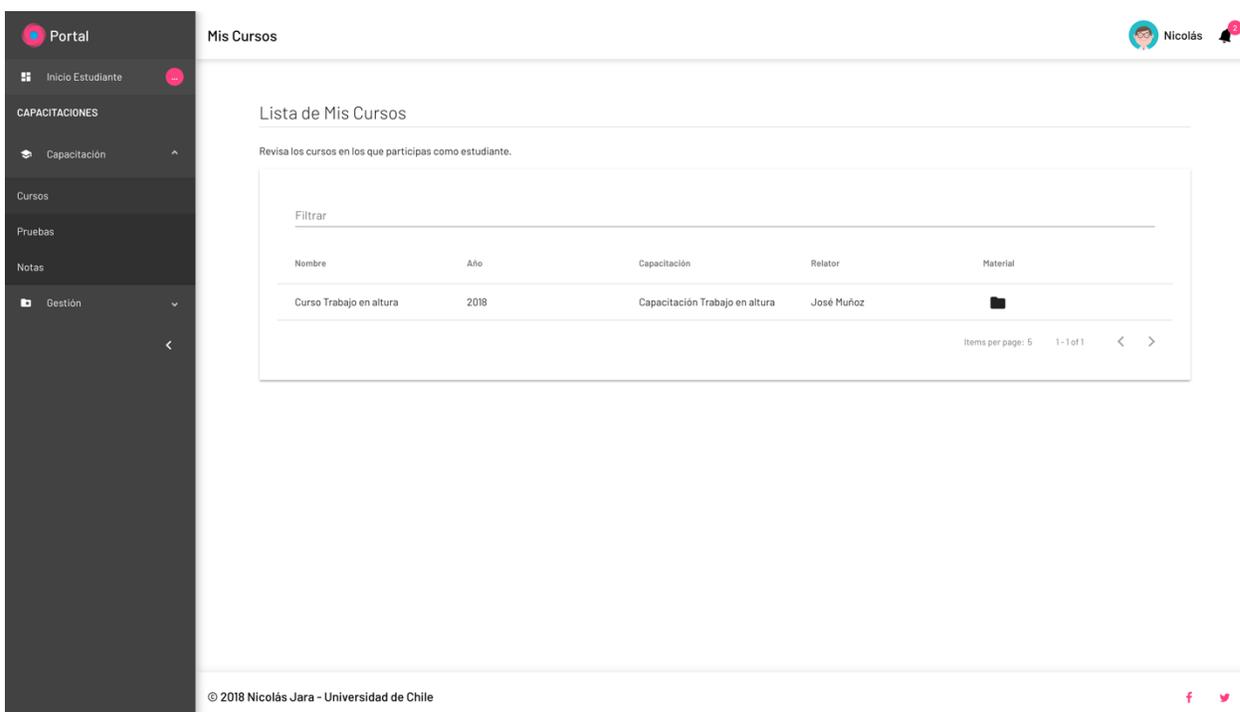


Figura 24. Vista de cursos en los que participa el estudiante.

4.4 INTERFACES DEL USUARIO EMPRESA

Las principales interfaces con las que el usuario empresa interactúa con el sistema, tienen relación con la administración de los alumnos (empleados de dicha empresa) y su documentación (diplomas y credenciales).

En primera instancia el usuario empresa debe iniciar sesión en el sistema con sus credenciales, y luego continuar revisando los estudiantes registrados en el sistema, que pertenecen a dicha empresa. Para ello debe dirigirse al menú lateral izquierdo donde se encuentra la sección "Usuarios Estudiantes", en donde visualizará una lista con los alumnos que están ejerciendo al menos una capacitación en el sistema (Figura 25).

Portal Nicolás

Inicio Administración

ADMINISTRACIÓN

Usuarios

Usuarios Administradores

Usuarios OTEC

Usuarios Empresas

Usuarios Relatores

Usuarios Estudiantes

Usuarios Técnicos

Direcciones

Agentes

Capacitación

Pruebas

Documentos

TÍTULO

Usuarios Estudiantes

Lista de Usuarios Estudiante

Crea, revisa y edita a los usuarios estudiantes en el sistema OTEC by Njara.

[Agregar Usuario](#)

Filtrar
sodimac

Rut ↑	Nombres	Apellidos	Email	Empresa	Ver / Editar
10011968	Martina	Burton	mburton@sodimac.cl	Sodimac	
100002043	Gustavo	Muñoz	alumno@eskipcapacitacion.cl	Sodimac	
100181525	Ernesto	Sierra	erelszroa@gmail.com	Sodimac	
100238421	Gloria	Gorro	glocfli_@hotmail.com	Sodimac	
100596301	José	Loyola	loyolaj7@gmail.com	Sodimac	

Items per page: 5 1 - 5 of 6

© 2018 Nicolás Jara - Universidad de Chile f t

Figura 25. Lista de estudiantes pertenecientes a la empresa.

5 EVALUACIÓN DE LA SOLUCIÓN

Dado que el trabajo de memoria se desarrolló en el marco de un OTEC particular, esto generó la oportunidad de poder validar el sistema web de forma temprana, siendo necesario ajustar en varias ocasiones esta solución. Los cambios realizados permitieron al OTEC mejorar la interacción con el sistema web.

Respecto a la mejora de los procesos del OTEC, recordemos que dentro de su gestión interna, el sistema de apoyo a estos procesos era usualmente una planilla Excel, con variadas funcionalidades (macros) que le permiten al OTEC automatizar cierto ingreso de información. En este punto, el sistema web desarrollado en esta memoria permitió, entre otras cosas, evitar los problemas de duplicación de información, dado que existe una validación de unicidad de los usuarios. Con esto se evita que los usuarios sean ingresados más de una vez al sistema. Lo mismo ocurre con los cursos y las capacitaciones; al existir un buscador, el OTEC puede revisar esta información sin tener que revisar planilla por planilla en su sistema de carpetas con planillas Excel.

Respecto a la modularidad de la solución, dentro del alcance no estaba considerado generar la automatización de credenciales; funcionalidad que fue solicitada por el OTEC durante el desarrollo de la solución, dado que al no poder trabajar en paralelo se generaron conflictos en la generación de éstas. Estos conflictos se daban al replicarse los identificadores de credenciales. Esta funcionalidad fue incorporada sin necesidad de modificar la arquitectura diseñada, permitiendo al OTEC generar credenciales automáticamente, mitigando el problema de generación de identificadores erróneos para las credenciales.

Finalmente, el OTEC presentó el interés de continuar el desarrollo del sistema más allá del alcance de este trabajo de memoria, con el fin de desarrollar de forma modular los procesos restantes para el apoyo completo de la operación del OTEC.

6 CONCLUSIONES Y TRABAJO A FUTURO

A medida que se diseñaba la arquitectura de la solución y el desarrollo de ella, proceso acompañado por el OTEC, se visualizaron los constantes cambios que experimentan estos organismos en la ejecución de capacitaciones y de sus procesos internos. Estos cambios se deben a la gran variedad de cursos que los OTECs proveen al mercado, por ejemplo: manejo de extintores, trabajo en altura, conducción de maquinaria pesada, entre otros. Cada capacitación tiene una forma propia de ser ejecutada, por ello, la solución durante su desarrollo necesitó consumir y representar estos cambios, debido a que el objetivo principal del trabajo de memoria era apoyar la gestión de los procesos internos y externos del OTEC. Dado este escenario, fue primordial el generar una arquitectura modular que permita acoplar y desacoplar funcionalidades según vayan cambiando estos procesos, permitiendo una constante evolución del sistema.

Respecto a las tecnologías utilizadas, se destaca la flexibilidad y utilidad que éstas prestaron durante el desarrollo de la solución, comprobando que el análisis previo de alternativas de soluciones y tecnologías a utilizar fue la correcta para este desarrollo.

Por otro lado, se generó una dinámica comercial con el software desarrollado, dado que no todos los OTEC poseen un sistema web para la gestión interna, con la posibilidad de darle acceso a sus clientes para la descarga de documentos, credenciales y/o diplomas. Esto permitió obtener un elemento diferenciador a la hora de presentar los servicios por parte del OTEC.

Finalmente, y dado los antecedentes previamente declarados, se hace obligadamente necesario mencionar que la solución aquí presentada no corresponde a la solución completa que necesita la OTEC para llevar a cabalidad sus procesos. Esto se debe al gran número de procesos que se realizan dentro y fuera del OTEC, que hacen imposible (por tamaño) desarrollar todo el sistema en el marco de un trabajo de memoria. Por ello, se definió como objetivo de esta memoria el contar con una arquitectura modular, que permita seguir desarrollando e incluyendo el soporte para nuevos procesos de cara al futuro. Algunos de los procesos que necesitan ser apoyados en los

distintos flujos de un OTEC, son los siguientes: seguimiento comercial de las capacitaciones, y generación de instancias de reportería. Este último incluye indicadores de aprobación, e indicadores de uso del sistema web, entre otros.

Más allá de la necesidad de contar con estos servicios extras, el OTEC que apoyó esta memoria quedó muy satisfecho con el trabajo realizado, indicando que la funcionalidad entregada ya le permite mejorar considerablemente su operación interna y externa, comparado con el uso de planillas Excel.

7 BIBLIOGRAFÍA

1. DEFINICIÓN SOFTWARE AS A SERVICE. [HTTP://WWW.INTERROUTE.ES/WHAT-SAAS](http://www.interoute.es/what-saas). ÚLTIMA VISITA: 10 DE SEP. DE 2018.
2. SERVICIO NACIONAL DE CAPACITACIÓN Y EMPLEO. [HTTP://WWW.SENCE.CL/PORTAL/PERFIL/ORGANISMOS/OTEC/](http://www.sence.cl/portal/perfil/organismos/otec/). ÚLTIMA VISITA: 10 DE SEP. DE 2018.
3. LISTADO ORGANISMOS TÉCNICOS DE CAPACITACIÓN (OTEC) VIGENTES. [HTTP://WWW.SENCE.CL/601/W3-PROPERTYVALUE-995.HTML?_NOREDIRECT=1](http://www.sence.cl/601/w3-propertyvalue-995.html?_noredirect=1). ÚLTIMA VISITA: 10 DE SEP. DE 2018.
4. LEY 19.518, FIJA NUEVO ESTATUTO DE CAPACITACIÓN Y EMPLEO. [HTTPS://WWW.LEYCHILE.CL/NAVEGAR?IDNORMA=76201](https://www.leychile.cl/Navegar?IDNorma=76201). ÚLTIMA VISITA: 10 DE SEP. DE 2018.
5. LEY 19.967, SOBRE RACIONALIZACIÓN DEL USO DE LA FRANQUICIA TRIBUTARIA DE CAPACITACIÓN. [HTTPS://WWW.LEYCHILE.CL/NAVEGAR?IDNORMA=229896](https://www.leychile.cl/Navegar?IDNorma=229896). ÚLTIMA VISITA: 10 DE SEP. DE 2018.
6. ANGULAR, FRAMEWORK PARA APLICACIONES WEB DE TYPESCRIPT DE CÓDIGO ABIERTO. [HTTPS://ANGULAR.IO/](https://angular.io/). ÚLTIMA VISITA: 10 DE SEP. DE 2018.
7. DJANGO, FRAMEWORK PARA APLICACIONES WEB GRATUITO Y OPEN SOURCE, ESCRITO EN PYTHON. [HTTPS://WWW.DJANGOPROJECT.COM](https://www.djangoproject.com)] ÚLTIMA VISITA: 10 DE SEP. DE 2018.
8. DJANGO REST FRAMEWORK, APLICACIÓN DJANGO PARA LA CREACIÓN DE WEB APIs. [HTTP://WWW.DJANGO-REST-FRAMEWORK.ORG/](http://www.django-rest-framework.org/). ÚLTIMA VISITA: 10 DE SEP. DE 2018.
9. RESOLUCIÓN EXENTA N°5477, ESTABLECE USO NUEVO LCE. [HTTP://WWW.SENCE.CL/601/ARTICLES-6671_ARCHIVO_01.PDF](http://www.sence.cl/601/articles-6671_archivo_01.pdf). ÚLTIMA VISITA: 10 DE SEP. DE 2018.

8 ANEXOS

8.A MANUAL DE INSTALACIÓN

Para completar esta guía de instalación se necesita una instancia de servidor Ubuntu 16.04 y un usuario con privilegios *sudo*. En este servidor se instalará el proyecto Django correspondiente al back-end y el proyecto Angular correspondiente al front-end. Luego se instalará y configurará el servidor de aplicaciones uWSGI quien jugará el rol de interfaz para las solicitudes HTTP que se realizarán a nuestra aplicación Django. Una vez tengamos nuestras aplicaciones desplegadas se instalará y configurará Nginx servidor web/proxy quién se encargará de ofrecer nuestra aplicación en la web.

8.A.1 Virtualenv y VirtualEnvWrapper

El proyecto Django se instalará en su propio ambiente esto para evitar conflictos si en la instancia del servidor se encuentran más aplicaciones por lo que con virtualenv se aíslan los requisitos de este proyecto en particular.

Instalaremos estos componentes con *pip* el administrador de paquetes para Python.

Escribir en consola:

```
$ sudo apt-get update
$ sudo apt-get install python3-pip
$ sudo apt-get install git
```

Teniendo pip instalado y git este último se utilizará para descargar el proyecto, queda instalar virtualenv y virtualenvwrapper, este último mejora la usabilidad de virtualenv.

```
$ sudo -H pip3 install --upgrade pip
$ sudo -H pip3 install virtualenv virtualenvwrapper
```

Con estos componentes instalados, se necesita configurar la shell para trabajar con virtualenvwrapper, ejecutar los siguientes comandos.

```
$ echo "export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3" >> ~/.bashrc
$ echo "export WORKON_HOME=~/.env" >> ~/.bashrc
$ echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.bashrc
$ source ~/.bashrc
```

con esto, deberíamos tener un directorio llamado Env en la carpeta de inicio.

8.A.2 Crear ambiente para el proyecto Django back-end y front-end.

Para crear el entorno virtual que aísle los requisitos de nuestro proyecto debemos ejecutar el siguiente comando:

```
$ mkvirtualenv otec
```

Esto creará nuestro entorno virtual otec, por lo que cualquier software que se instale desde ahora, se instalará en este entorno.

Teniendo el entorno virtual queda instalar los requisitos del proyecto, partiendo con la instalación de Django.

```
$ sudo pip3 install Django==2.0.3
```

Ahora queda descargar el proyecto desde bitbucket donde se encuentra alojado o traspararlo al servidor con cualquier cliente FTP que permita transferir los archivos al servidor como por ejemplo, Filezilla. El proyecto lo dejaremos en el directorio `~/home/ubuntu/backend/`.

```
$ git clone https://nicolasjara@bitbucket.org/memoriaotec/backend-otec.git
```

Teniendo el proyecto, se debe ejecutar la instalación de los requisitos del proyecto a través del archivo `requirements.txt`, en el cual se detalla cada

```
$ pip3 install -r requirements.txt
```

requisito con la versión específica.

Ahora quedan los comandos para publicar el proyecto Django.

```
(otec) $ cd ~/otec
(otec) $ ~/otec/manage.py migrate
(otec) $ ~/otec/manage.py createsuperuser
(otec) $ ~/otec/manage.py collectstatic
```

Estos comandos gestionan los archivos que se desplegarán del proyecto back-end.

Respecto al proyecto front-end, dentro del proyecto debemos ejecutar el siguiente comando que compilará los archivos productivos del mismo.

```
$ build --prod
```

Esto generará la carpeta "dist", en la cual encontraremos todos los archivos que debemos disponer en internet que corresponden al proyecto front-end.

En nuestro caso, los dejaremos en el directorio "/home/ubuntu/frontend/".

Para continuar con los siguientes pasos a configurar se debe salir del entorno virtual usando el siguiente comando:

```
$ deactivate
```

8.A.3 Instalación y configuración de uWSGI

Estando fuera de nuestro entorno virtual, debemos instalar uWSGI globalmente.

```
$ sudo -H pip3 install uwsgi
```

Teniendo instalado uWSGI necesitamos generar el directorio donde se gestionarán el o los proyectos Django que tendremos en el servidor.

```
$ sudo mkdir -p /etc/uwsgi/sites
```

En este directorio crearemos el archivo de configuración para nuestro proyecto Django back-end,

```
$ sudo nano /etc/uwsgi/sites/otec.ini
```

en el copiaremos la siguiente configuración que corresponde a nuestro proyecto:

```
[uwsgi]
project = otec
uid = ubuntu
base = /home/%(uid)

chdir = %(base)/%(project)
home = %(base)/Env/%(project)
module = %(project).wsgi:application

master = true
processes = 5

socket = /run/uwsgi/%(project).sock
chown-socket = %(uid):www-data
chmod-socket = 660
vacuum = true
```

8.A.4 Automatizar la ejecución de uWSGI

A continuación, crearemos un archivo de *systemd* para administrar el proceso de uWSGI para que se inicie al iniciar el sistema.

Para ello crearemos un archivo en `"/etc/systemd/system/"`.

```
$ sudo nano /etc/systemd/system/uwsgi.service
```

Y copiaremos el siguiente código que corresponde al de nuestro proyecto.

```
[Unit]
Description=uWSGI Emperor service

[Service]
ExecStartPre=/bin/bash -c 'mkdir -p /run/uwsgi; chown ubuntu:www-data /run/uwsgi'
ExecStart=/usr/local/bin/uwsgi --emperor /etc/uwsgi/sites

Restart=always
KillSignal=SIGQUIT
Type=notify
NotifyAccess=all

[Install]
WantedBy=multi-user.target
```

Para iniciar este servicio aún nos falta instalar y configurar Nginx ya que depende de éste.

8.A.5 Instalación y configuración de Nginx como proxy inverso.

Ya teniendo nuestro servicio de uWSGI instalado y configurado nos queda instalar y configurar Nginx como un proxy inverso.

Para instalarlo debemos ejecutar el siguiente comando:

```
$ sudo apt-get install nginx
```

Ya instalado, debemos crear el archivo de configuración.

```
$ sudo nano /etc/nginx/sites-available/frontend
```

Y agregar el siguiente código que corresponde a nuestra configuración del proyecto (cambiar la variable {IP_SERVER}, por la ip del servidor donde se está ejecutando este manual).

```
server {  
    listen 80;  
    server_name {IP_SERVER};  
  
    root /home/ubuntu/frontend/;  
    index index.html;  
  
    location / {  
        try_files $uri $uri/ /index.html;  
    }  
  
    # Handles proxying to Django API  
    location /otec {  
        include uwsgi_params;  
        uwsgi_pass unix:/run/uwsgi/otec.sock;  
    }  
}
```

Teniendo esta configuración, continua habilitar el sitio para que sea accesible.

Para ello, ejecutamos:

```
$ sudo ln -s /etc/nginx/sites-available/frontend /etc/nginx/sites-enabled
```

Comprobamos la sintaxis:

```
$ sudo nginx -t
```

Si todo está ok, reiniciamos el servicio de Nginx:

```
$ sudo systemctl restart nginx
```

Como ya está configurado e iniciado el servicio de Nginx, podemos continuar con el paso pendiente de la sección 8.1.4:

```
$ sudo systemctl start uwsgi
```

Luego generamos los permisos del puerto 8080, para que sea utilizado por Nginx:

Queda entonces habilitar que estos servicios de uWSGI y Nginx inicien automáticamente al arrancar el sistema:

```
$ sudo ufw delete allow 8080  
$ sudo ufw allow 'Nginx Full'
```

```
$ sudo systemctl enable nginx  
$ sudo systemctl enable uwsgi
```

Ya tenemos nuestros proyectos disponibles en un servidor. Basta con ingresar a un navegador de internet y digitar la IP del servidor para visualizar la pantalla de Iniciar Sesión de nuestro sistema de apoyo a la gestión de un OTEC.

8.A.6 Modificaciones a los proyectos back-end y front-end.

Si se realizan modificaciones al proyecto front-end, se debe volver a compilar con el comando:

```
$ build --prod
```

Y reemplazar los archivos por los anteriores y reiniciar los servicios de uWSGI y Nginx con los comandos:

```
$ sudo service uwsgi restart  
$ sudo service nginx restart
```

En el caso de que las modificaciones se hayan realizado en el proyecto-backend, se deben repetir los pasos de descargar el proyecto nuevamente (o

```
(otec) $ cd ~/otec  
(otec) $ ~/otec/manage.py migrate  
(otec) $ ~/otec/manage.py collectstatic
```

con GIT, solo descargar los cambios con “pull”) y ejecutar los siguientes comandos:

Y reiniciar los servicios de uWSGI y Nginx.

```
$ sudo service uwsgi restart  
$ sudo service nginx restart
```

8.B DOCUMENTACIÓN API PROYECTO BACK-END.

La documentación de la API correspondiente al proyecto back-end fue generada automáticamente por el software Postman, y publicada en la siguiente url:

<https://documenter.getpostman.com/view/888747/RWaGVVPU#40edd6e1-9628-4ed8-b258-586cb110227f>

La documentación consta de las principales url's que consume el proyecto front-end. En ella se detalla la url del servicio, el request y el response del mismo. Esta documentación además permite comprobar el servicio en distintos lenguajes de programación.