**Sistema de pre inscripción Institutos Superiores**

* Nombre de la aplicación:
* Descripción: Sistema de Preinscripción y Gestión Académica para Institutos de Educación Superior en la Provincia de Buenos Aires

**Índice:**

1. Introducción
2. Objetivos
3. Arquitectura del Sistema
4. Funcionalidades Principales
   * Regional
   * Local
5. Tecnologías Utilizadas
6. Análisis de fallas
   1. Errores de navegación y re direccionamiento
   2. Errores de sesión
   3. Errores de navegación dentro del sistema
   4. Errores de acceso a funcionalidades
   5. Errores de lógica en la preinscripción
   6. Errores en la validación de datos
   7. Mejoras en la interfaz de usuario
   8. Mejoras en seguridad
   9. Errores en el flujo de la aplicación
7. Modificaciones en el Diagrama de Entidad de Relación (DER)
8. Plan de Correcciones

**1. Introducción:**

* Este sistema surge como respuesta a la necesidad de simplificar y mejorar el proceso de preinscripción y gestión académica para estudiantes de institutos de educación superior en la Provincia de Buenos Aires. Con el objetivo de facilitar el acceso a la educación superior y optimizar la administración académica.

**2. Objetivos:**

* Generales:
  + Facilitar el proceso de preinscripción a carreras terciarias de los institutos superiores de la Provincia de Buenos Aires.
  + Centralizar la gestión académica para institutos de educación superior.
* Específicos:
  + Brindar a los estudiantes información detallada sobre las carreras disponibles, incluyendo modalidades y horarios.
  + Permitir a los institutos gestionar eficientemente la inscripción a materias, finales y el seguimiento del rendimiento académico de los Estudiantes.

**3. Arquitectura del Sistema:**

* “Nombre del sistema” está diseñado como un sistema web modular basado en Symfony, un framework PHP robusto y flexible.
* La arquitectura consta de dos partes principales:
  + **Base Regional:** Donde los estudiantes pueden realizar preinscripciones y acceder a información sobre las carreras ofrecidas por los institutos, modalidad y horarios.
  + **Base Local:** Plataforma privada para cada instituto, donde se gestiona la inscripción a materias, se realizan seguimientos de notas, etc.

**4. Funcionalidades Principales:**

**Base Regional:**

* **Preinscripción a Carreras:** Los estudiantes pueden registrarse y preinscribirse a las carreras de su interés.
* **Información de Carreras:** Acceso a datos sobre modalidades, horarios y oferta educativa de los institutos.

**Base Local:**

* **Gestión de Estudiantes:** Los institutos pueden administrar las inscripciones de los estudiantes.
* **Inscripción a Materias y Finales:** Los estudiantes pueden seleccionar y registrar las materias que desean cursar y los finales que desean rendir.
* **Seguimiento Académico:** Herramientas para monitorear el desempeño académico de los estudiantes, incluyendo registro de notas, asistencia, y cualquier otra métrica relevante para evaluar su progreso académico.

**5. Tecnologías Utilizadas:**

* **Symfony:** Framework PHP utilizado para el desarrollo del sistema, proporcionando una estructura sólida y flexible.
* **PHPMyAdmin:** Herramienta de administración de base de datos utilizada para gestionar la base de datos relacional del sistema.
* **Composer:** Gestor de dependencias utilizado para manejar las bibliotecas y paquetes PHP necesarios para el proyecto.
* **Visual Studio Code:** entorno de desarrollo integrado (IDE) ampliamente utilizado por programadores de software para escribir, editar y depurar código.

Algunas de sus características principales incluyen resaltado de sintaxis, autocompletado inteligente, depuración integrada, control de versiones con Git, soporte para extensiones personalizadas, administración de paquetes, y una interfaz de usuario altamente personalizable.

**6 Análisis de Fallas**

Realizamos un análisis exhaustivo de fallas en el programa desarrollado. Durante este análisis, identificamos varios errores y problemas que afectan la funcionalidad y la experiencia del usuario.

Algunos de los problemas detectados incluyen problemas de validación de datos en el formulario de preinscripción, inconsistencias en la interfaz de usuario, errores de lógica en el proceso de almacenamiento y recuperación de datos, y falta de manejo de errores y mensajes de error poco claros para los usuarios.

**Preinscripción Regional**

A continuación, se detallan los errores detectados a solucionar a corto plazo

**1. Errores de navegación y redireccionamiento:**

* Routear al home si no se loguea como usuario admin.
* Error al intentar acceder por otra ruta que no sea el home.

**2. Errores de sesión:**

* Dificultad para ingresar, generalmente se logra en el segundo intento.
* Al cerrar sesión y regresar con la tecla “retroceso” o “backspace”, se reingresa al sistema.

**3. Errores de navegación dentro del sistema:**

* Al ingresar al sistema no se routea al home o index.

**4. Errores de acceso a funcionalidades:**

* No se puede acceder al formulario de preinscripción estando logueado.
* El acceso manual al formulario de preinscripción genera un error.

**5. Errores en la lógica de la preinscripción:**

* El orden para la preinscripción parece ser aleatorio en lugar de seguir un orden específico.
* Permitir la carga de DNI idénticos, lo que duplicaría una preinscripción.
* Permitir la preinscripción con campos de datos incompletos.
* Falta incluir tecnicaturas en las opciones de preinscripción.
* Falta cargar institutos en las opciones de preinscripción.
* Cambiar "alumno" por "estudiante" en la interfaz de usuario.
* La fecha de preinscripción debería ser automática.

**6. Errores en la validación de datos:**

* En la carga de datos personales, los campos no validan la entrada correctamente.
* Permitir la carga de caracteres incorrectos en campos específicos (por ejemplo, números en el campo de nombre).

**7. Mejoras en la interfaz de usuario:**

* Algunos campos deberían ser combobox para mejorar la experiencia de usuario.
* Falta ampliar las opciones de carga en ciertos campos (por ejemplo, localidad).
* Agregar más campos en la sección de género y localidad.

**8. Mejoras en la seguridad:**

* Falta algún tipo de validación de preinscripción (por ejemplo, confirmación a través de correo electrónico).
* Falta implementar medidas de seguridad para verificar la identidad del usuario durante la preinscripción.

**9. Errores en el flujo de la aplicación:**

* Al presionar la tecla "Enter" en uno de los campos de datos personales, carga la preinscripción automáticamente sin completar los datos.
* Como admin no está permitiendo borrar una preinscripción.
* Falta un botón de cancelar preinscripción.

*El análisis detallado del programa ha evidenciado una serie de errores y deficiencias que requieren atención inmediata para mejorar la experiencia del usuario y garantizar el correcto funcionamiento del sistema. Los problemas identificados abarcan diversas áreas, desde errores de navegación y re direccionamiento hasta fallos en la lógica de la preinscripción y deficiencias en la validación de datos.*

*Se deben implementar medidas para corregir los errores de navegación y re direccionamiento, mejorar la lógica de la preinscripción para seguir un orden específico y garantizar que los datos ingresados sean válidos y completos.*

*Además, se deben realizar mejoras en la interfaz de usuario, como la implementación de combobox en campos relevantes y la ampliación de opciones de carga para mejorar la experiencia del usuario. Asimismo, es de importancia fortalecer la seguridad del sistema mediante la implementación de medidas de verificación de identidad y validación de preinscripciones.*

* 1. **Modificaciones en el Diagrama de Entidad de Relación (DER)**

Para poder llevar dar continuidad al proyecto en Symfony, hemos realizado una serie de modificaciones importantes en lo que al diagrama de entidad de relación que a continuación se detallan:

* Agregado del atributo "clave" a la entidad Instituto:

La entidad Instituto no contaba con un atributo único que la identificara de manera específica. Por esta razón, se añadió el atributo "clave" para mejorar la identificación y el manejo de los datos relacionados a los institutos.

Creación de la entidad Distrito y establecimiento de una relación 1 a N con la entidad Localidad:

* Se creó una nueva entidad denominada Distrito para gestionar mejor la organización regional.

Se estableció una relación uno a muchos entre la entidad Distrito y la entidad Localidad. Esto implica que cada distrito puede tener múltiples localidades asociadas.

Para implementar esta relación, se agregó el atributo distrito\_id en la entidad Localidad como clave foránea (FK), permitiendo así vincular cada localidad con su correspondiente distrito.

* Actualización de la base de datos regional:

Con las nuevas entidades y atributos, fue necesario actualizar la estructura de la base de datos regional.

Se realizaron las migraciones correspondientes para reflejar estos cambios y asegurar la integridad de los datos.

* Creación de clases Repository, Controller y Form (Type):

Se crearon las clases repository necesarias para gestionar las operaciones de persistencia de datos de las nuevas entidades y atributos.

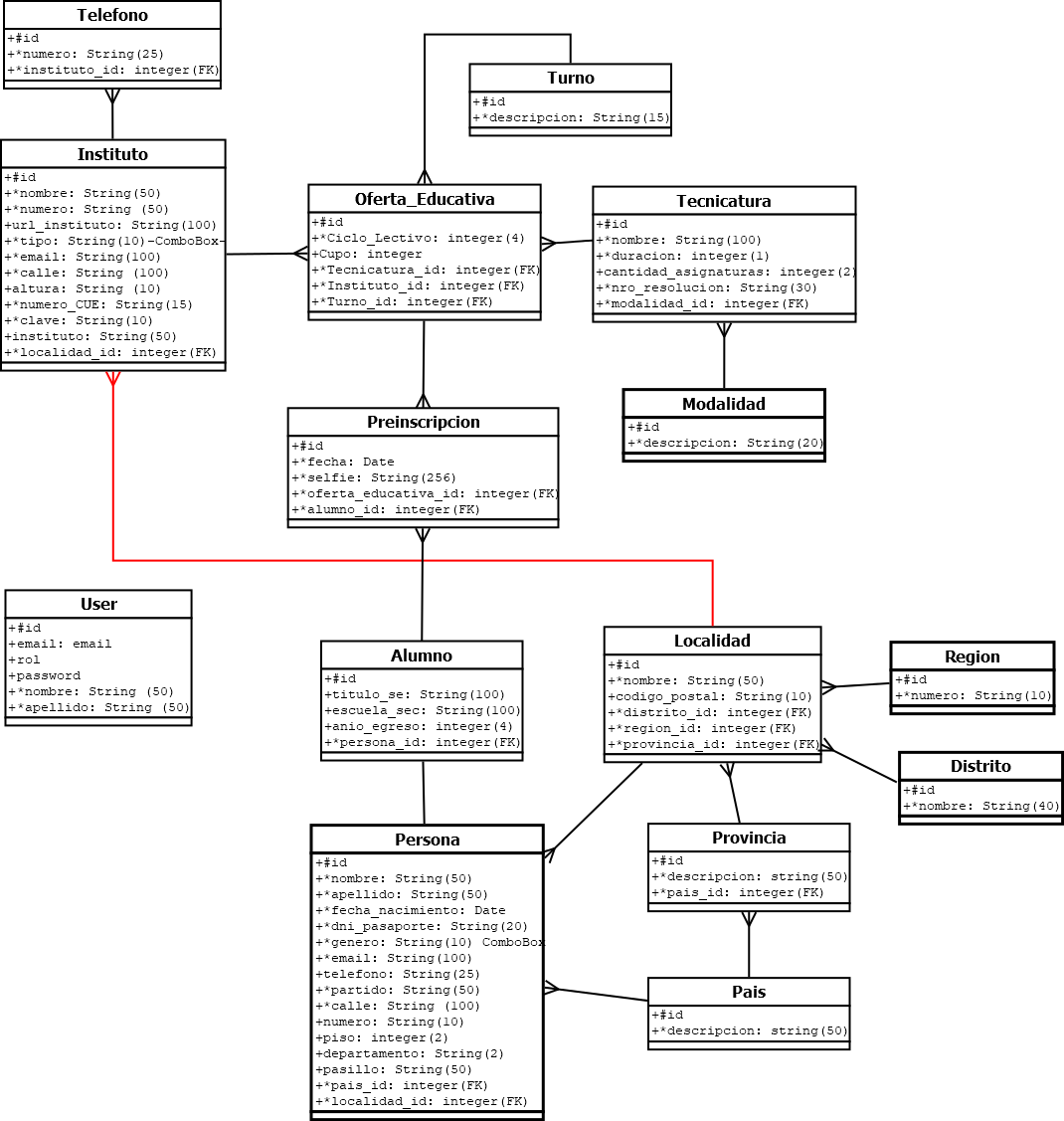
* Se implementaron las clases controller correspondientes para manejar las solicitudes y respuestas relacionadas con las entidades Instituto y Distrito.
* Se desarrollaron las clases Form (Type) para facilitar la creación y gestión de formularios en la aplicación.
* Actualización de formulario (InstitutoType) y templates:

El formulario InstitutoType se actualizó para incluir el nuevo atributo "clave", permitiendo así que este dato sea ingresado y gestionado a través de la interfaz de usuario.

* Los templates correspondientes se modificaron para reflejar estos cambios y asegurar que la nueva funcionalidad sea accesible y utilizable por los usuarios del sistema.

*Estas modificaciones fueron esenciales para completar y mejorar el proyecto, asegurando que las nuevas entidades y relaciones se integren de manera efectiva en la aplicación. Con estos cambios, se logró una mejor organización y manejo de los datos regionales, así como una mayor funcionalidad y usabilidad en el sistema*

**Nuevo Diagrama de Entidad de Relacion (DER)**

****

* 1. **Plan de Correcciones:**

En respuesta a los desafíos identificados en nuestra aplicación, estamos llevando a cabo una investigación exhaustiva para identificar las mejores soluciones utilizando las herramientas proporcionadas por Symfony, incluyendo Symfony Routing, Symfony Security Component y Symfony Forms, entre otras.

**1. Symfony Routing:**

Una de las áreas clave que estamos investigando es el enrutamiento dentro de nuestra aplicación. Estamos evaluando cómo podemos configurar correctamente las rutas para redirigir a los usuarios a las páginas adecuadas, especialmente en casos como el acceso no autorizado o el re direccionamiento después del cierre de sesión. Utilizaremos Symfony Routing para definir y gestionar estas rutas de manera eficiente.

**2. Symfony Security Component:**

La seguridad es una prioridad fundamental para nosotros. Estamos explorando cómo podemos utilizar el Symfony Security Component para implementar autenticación, autorización y control de acceso en nuestra aplicación. Esto incluye garantizar que solo los usuarios autorizados puedan acceder a ciertas funcionalidades y que se implementen medidas adecuadas para proteger los datos sensibles.

**3. Symfony Forms:**

Para mejorar la experiencia del usuario y garantizar la integridad de los datos ingresados, estamos considerando cómo podemos utilizar Symfony Forms para crear y validar formularios de manera efectiva. Esto incluye garantizar que los datos ingresados por los usuarios cumplan con ciertos criterios y que se presenten de manera clara y coherente en la interfaz de usuario.

**4. Symfony Twig:**

Además de Symfony Forms, también estamos evaluando cómo podemos utilizar Symfony Twig para gestionar y renderizar las vistas de nuestra aplicación de manera eficiente. Esto incluye la creación de plantillas reutilizables y la implementación de prácticas recomendadas para mejorar la legibilidad y el mantenimiento del código.

**5. Pruebas y Validación:**

Una vez que hayamos identificado las soluciones adecuadas utilizando las herramientas de Symfony, realizaremos pruebas exhaustivas para garantizar su eficacia y fiabilidad. Esto incluye pruebas de unidad, pruebas de integración y pruebas de usuario para asegurarnos de que las correcciones implementadas cumplan con nuestras expectativas y requisitos.

**1. Errores de navegación y re direccionamiento:**

* **Symfony Routing:** Aseguraremos que las rutas redirijan adecuadamente al home si el usuario no está logueado como admin y corregiremos cualquier problema con las rutas que no dirijan correctamente al home.

**2. Errores de sesión:**

* **Symfony Security Component:** Investigaremos y solucionaremos los problemas de inicio de sesión, así como implementaremos un manejo adecuado de sesiones para evitar que el usuario vuelva a ingresar al sistema después de cerrar sesión.
  1. **Errores de navegación dentro del sistema:**
* **Symfony Routing:** Verificaremos que todas las rutas estén correctamente configuradas para dirigir al usuario al home o index después del inicio de sesión.
  1. **Errores de acceso a funcionalidades:**
* **Symfony Security Component:** Aseguraremos que las restricciones de acceso estén configuradas correctamente para que el formulario de preinscripción no sea accesible para usuarios logueados, además de corregir cualquier problema con el acceso manual al formulario de preinscripción.

**5. Errores en la lógica de la preinscripción:**

* **Symfony Forms:** Actualizaremos la lógica del formulario para garantizar que el orden de la preinscripción sea correcto y que no se permita la carga de DNI idénticos o campos de datos incompletos.
* **Base de datos:** Agregaremos las tecnicaturas y los institutos faltantes a la base de datos.
* **Interfaz de usuario:** Modificaremos la interfaz para cambiar "alumno" por "estudiante" y aseguraremos que la fecha de preinscripción se genere automáticamente.

**6. Errores en la validación de datos:**

* **Symfony Forms:** Implementaremos validaciones adecuadas en los campos del formulario para garantizar la entrada correcta de datos.
* **Validación de Datos:** Realizaremos validaciones adicionales a nivel de servidor si es necesario para prevenir la carga de caracteres incorrectos.

**7. Mejoras en la interfaz de usuario:**

* **Symfony Forms / Twig:** Convertiremos algunos campos en combobox para mejorar la experiencia del usuario.
* **Interfaz de usuario:** Ampliaremos las opciones de carga en los campos de localidad y género según sea necesario.

**8. Mejoras en la seguridad:**

* **Symfony Security Component:** Implementaremos la validación de preinscripción mediante correo electrónico y verificaremos la identidad del usuario durante el proceso de preinscripción.
* **Seguridad de la Aplicación:** Añadiremos medidas de seguridad adicionales según las mejores prácticas de Symfony y la seguridad web en general.

**9. Errores en el flujo de la aplicación:**

* **Symfony Forms / JavaScript:** Evitaremos la carga automática de preinscripción al presionar "Enter" en un campo de datos personales.
* **Administración de Preinscripciones:** Solucionaremos el problema que impide que el administrador borre una preinscripción.
* **Interfaz de usuario:** Agregaremos un botón de cancelar preinscripción para mejorar el flujo de la aplicación.

Este plan aborda las correcciones necesarias en cada área utilizando las tecnologías mencionadas, abordando así los problemas identificados y mejorando la experiencia del usuario y la seguridad del sistema.