com.cdsoftware.appointment

Documentación técnica del plugin iDempiere 12 para gestión reutilizable de citas, servicios, empleados, horarios, disponibilidad, generación de slots y validación centralizada desde UI o API.

- Visión General
  - Resumen del Plugin
  - Arquitectura y Componentes
  - Flujo de Configuración y Cita
- Modelo de Datos
  - Tablas y Modelo de Datos
- Lógica Backend
  - Motor de Disponibilidad
  - Callouts y Flujo de UI
  - Procesos y Eventos
- Operación e Integración
  - Uso desde REST/API
  - Instalación y 2Pack

Visión General

Resumen funcional y arquitectura general del plugin de citas.

Resumen del Plugin

El plugin `com.cdsoftware.appointment` agrega a iDempiere 12 un módulo reutilizable para gestionar citas, servicios, empleados, horarios, días no disponibles y disponibilidad de agenda.

> **Idea clave:** la disponibilidad no depende solo de la interfaz. La validación queda centralizada en backend para proteger tanto las citas creadas desde ventanas iDempiere como las creadas desde REST/API.

Objetivo

El objetivo es permitir que una organización configure servicios agendables, defina qué empleados pueden atender cada servicio, establezca horarios de atención y bloquee días o rangos no disponibles. Con esa información, el sistema genera slots disponibles y valida cada cita antes de guardarla.

Alcance funcional

Área	Qué cubre
Configuración	Parámetros generales por organización, límites de anticipación, reglas de aprobación e intervalos de slots.
Servicios	Servicios agendables con precio, duración, intervalo, capacidad y requerimiento de aprobación.
Empleados	Personas o usuarios disponibles para atender citas, vinculados a terceros y opcionalmente a usuarios iDempiere.
Disponibilidad	Horarios semanales, días libres de empleado, días libres de organización y validación de conflictos.

Área	Qué cubre
Citas	Registro principal de la cita con cliente, empleado, servicio, fecha, hora, duración, precio, estado y slot seleccionado.

Principios de diseño

- El plugin es genérico y no depende del proyecto IEA.
- La creación desde REST/API usa la tabla estándar `CDS_Appointment`.
- Las reglas críticas se validan en eventos de modelo, no solo en callouts.
- Los modelos generados `I_CDS_*` y `X_CDS_*` son la fuente de constantes de tabla, columnas y valores.
- El campo `R_Request_ID` puede existir como referencia manual, pero el código del plugin no depende de solicitudes.

Arquitectura y Componentes

La arquitectura del plugin separa la definición de modelos, la disponibilidad, la experiencia de ventana iDempiere y la validación obligatoria antes de guardar citas.

Componentes principales

Componente	Responsabilidad
`AppointmentAvailabilityEngine`	Centraliza cálculo de disponibilidad, validación de horarios, días libres, conflictos y generación de slots.
`AppointmentModelValidator`	Valida citas antes de crear o modificar registros, incluyendo consistencia del slot y conflictos de agenda.
`CalloutAppointment`	Apoya la captura en ventana iDempiere copiando precio, duración, empleado y horas desde servicio o slot.
`GenerateAppointmentSlots`	Genera registros `CDS_AppointmentSlot` para un empleado, servicio y fecha específicos.
`ReopenAppointment`	Permite reabrir una cita procesada de forma controlada.

Flujo general

1. El usuario configura servicios, empleados, capacidades, horarios y días no disponibles.
2. Desde una cita se selecciona empleado, servicio y fecha.
3. El proceso de generación crea slots disponibles para esa combinación.
4. El usuario selecciona un slot y el callout copia datos a la cita.
5. Al guardar, el evento valida disponibilidad, consistencia y conflictos.
6. Los clientes REST/API pueden crear la cita directamente y reciben la misma validación por evento.

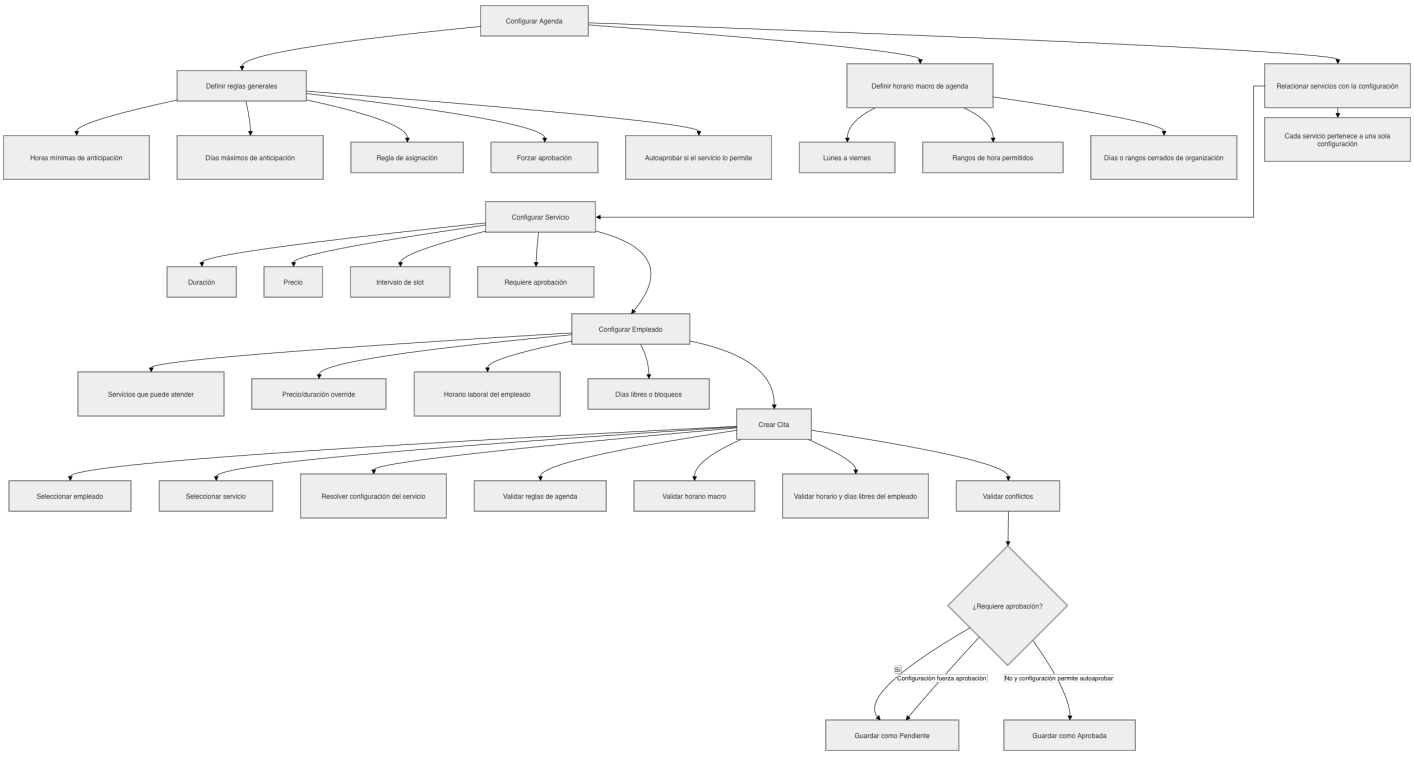
> **Nota:** los callouts mejoran la experiencia de captura, pero no son la capa de seguridad. La regla final vive en `AppointmentModelValidator`.

Flujo de Configuración y Cita

Esta página describe el flujo objetivo para preparar una agenda y poder crear citas válidas. El flujo parte de una configuración general, continúa con servicios y empleados, y termina con la creación de una cita validada por backend.

> **Importante:** parte de este flujo representa el comportamiento objetivo. Algunas reglas todavía deben implementarse en el modelo y en `AppointmentModelValidator` para que sean obligatorias tanto desde la UI como desde REST/API.

Flujo objetivo



Secuencia funcional

1. Crear una configuración de agenda por organización o por alcance operativo.

2. Definir reglas generales como anticipación mínima, anticipación máxima, regla de asignación y política de aprobación.
3. Definir el horario macro de la agenda. Por ejemplo: lunes a viernes de 8:00 a 17:00.
4. Relacionar cada servicio con una sola configuración de agenda.
5. Configurar cada servicio con duración, precio, intervalo de slot y si requiere aprobación.
6. Configurar empleados y definir qué servicios puede atender cada uno.
7. Definir horario laboral y días libres del empleado.
8. Crear la cita seleccionando empleado, servicio, fecha y slot.
9. Validar primero la configuración de agenda y luego las reglas del empleado.
10. Guardar la cita como pendiente o aprobada según las reglas de aprobación.

Jerarquía de disponibilidad esperada

Nivel	Regla	Prioridad
Configuración de agenda	Define el marco general de operación: anticipación, días permitidos, horarios macro y aprobación.	Debe estar por encima del empleado. Si la agenda general está cerrada, ningún empleado puede abrir disponibilidad.
Servicio	Define duración, precio, intervalo y si requiere aprobación.	Debe resolver qué configuración aplica.
Empleado	Define servicios que puede atender, horario propio y bloqueos personales.	Solo puede reducir o especializar la disponibilidad de la configuración, no ampliarla.
Cita	Consume empleado, servicio, configuración, slot y reglas de conflicto.	Debe ser validada por evento antes de guardarse.

Reglas pendientes para completar el flujo

- Relacionar cada servicio con una configuración de agenda.
- Validar que un servicio no pertenezca a más de una configuración.
- Agregar horario macro de configuración, similar al horario del empleado.

- Validar anticipación mínima y máxima desde la configuración.
- Aplicar la política de aprobación automática o forzada en `BeforeNew`.
- Hacer que los días libres de organización puedan aplicar por configuración cuando corresponda.

Modelo de Datos

Tablas, modelos generados y relaciones principales del módulo.

Tablas y Modelo de Datos

El 2Pack del plugin define las tablas del módulo y genera modelos Java `I_CDS_*` y `X_CDS_*`. El código usa esos modelos como fuente de nombres de tabla, columnas y valores de referencia.

Tablas principales

Tabla	Uso
`CDS_AppointmentConfig`	Configuración organizacional de agenda, reglas de asignación, aprobación, límites e intervalos.
`CDS_AppointmentService`	Servicios agendables con precio, duración, intervalo de slot, aprobación requerida y capacidad.
`CDS_AppointmentEmployee`	Empleado disponible para citas, vinculado a `C_BPartner` y opcionalmente a `AD_User`.
`CDS_EmployeeService`	Relación empleado-servicio con posibles overrides de precio y duración.
`CDS_EmployeeScheduleLine`	Bloques semanales de disponibilidad del empleado por día y rango horario.
`CDS_EmployeeDayOff`	Bloqueos de empleado por día completo o rango horario, incluyendo recurrencia anual.
`CDS_OrgDayOff`	Bloqueos de organización por día completo o rango horario. Incluye `CDS_AppointmentConfig_ID`, aunque el motor actual todavía no filtra por configuración.
`CDS_AppointmentSlot`	Slots generados para selección desde la ventana iDempiere y consulta desde API.
`CDS_Appointment`	Registro principal de cita con cliente, empleado, servicio, fecha, hora, precio, duración, estado y slot.

Relaciones clave actuales

- `CDS_Appointment` referencia cliente, empleado, servicio y opcionalmente slot.
- `CDS_AppointmentSlot` se genera para un empleado, servicio, fecha y rango horario.
- `CDS_EmployeeService` determina si un empleado puede prestar un servicio.
- `CDS_EmployeeScheduleLine` define cuándo puede atender el empleado.
- `CDS_EmployeeDayOff` y `CDS_OrgDayOff` bloquean disponibilidad.

Tabla `CDS_AppointmentSlot`

`CDS_AppointmentSlot` almacena horarios generados por usuario para que la UI o un cliente API pueda listar opciones disponibles después de ejecutar `GenerateAppointmentSlots`.

Columna	Uso	Filtro API recomendado
`CDS_AppointmentSlot_ID`	Identificador del slot seleccionado.	Usar al copiar el slot hacia `CDS_Appointment.CDS_AppointmentSlot_ID`.
`AD_Org_ID`	Organización del slot.	Filtrar por organización activa del usuario.
`AD_User_ID`	Usuario que generó el slot.	Filtrar por usuario actual para no mezclar slots temporales de otros usuarios.
`CDS_Appointment_ID`	Cita desde la cual se generó el slot, si aplica.	Filtrar cuando los slots se generaron desde una cita específica.
`CDS_AppointmentEmployee_ID`	Empleado disponible para atender el slot.	Filtrar por empleado seleccionado.
`CDS_AppointmentService_ID`	Servicio del slot.	Filtrar por servicio seleccionado.
`SlotDate`	Fecha del slot normalizada al inicio del día.	Filtrar por fecha de la cita.
`StartDateTime`	Inicio exacto del horario disponible.	Ordenar ascendente o filtrar por rango horario.
`EndDateTime`	Fin exacto del horario disponible.	Usar junto con `StartDateTime` para mostrar rango.

Columna	Uso	Filtro API recomendado
`DisplayName`	Etiqueta legible del horario.	Mostrar como texto de selección.
`IsAvailable`	Indica si el slot sigue disponible.	Filtrar `IsAvailable='Y'`.
`IsActive`	Control estándar de registro activo.	Filtrar `IsActive='Y'`.
`AvailableEmployees`	Cantidad de empleados disponibles para el rango. En el flujo actual por empleado suele quedar en `1`.	Usar solo como referencia informativa.
`Price`	Precio efectivo del slot.	Copiar a la cita si se selecciona el slot.
`DurationMinutes`	Duración efectiva del slot.	Copiar a la cita si se selecciona el slot.
`AssignedByRule`	Indica si el empleado fue asignado por regla automática.	Actualmente informativo en el flujo por empleado.
`AssignmentRuleUsed`	Regla de asignación usada al generar el slot.	Usar para auditoría o depuración.

> **Consulta típica API:** después de generar slots, listar `CDS_AppointmentSlot` filtrando por `AD_User_ID`, `CDS_AppointmentEmployee_ID`, `CDS_AppointmentService_ID`, `SlotDate`, `IsActive='Y'` e `IsAvailable='Y'`, ordenando por `StartDateTime`.

Relación configuración-servicio pendiente

El flujo objetivo requiere que cada servicio resuelva una configuración de agenda. Esto permite saber qué reglas de anticipación, horarios macro, días cerrados y aprobación aplican cuando se crea una cita.

Necesidad	Opción recomendada	Validación requerida
Servicio pertenece a una configuración	Agregar `CDS_AppointmentConfig_ID` en `CDS_AppointmentService`.	El servicio debe tener una configuración activa cuando se use para generar slots o crear citas.

Necesidad	Opción recomendada	Validación requerida
Un servicio no puede tener reglas ambiguas	Si se usa tabla intermedia, imponer unicidad por servicio activo.	Bloquear más de una configuración activa por servicio.
Días libres por configuración	Usar el campo existente `CDS_AppointmentConfig_ID` de `CDS_OrgDayOff`.	El motor debe filtrar bloqueos por organización y por configuración del servicio.

Horario macro de configuración pendiente

Actualmente solo existe horario de empleado. Para cumplir el flujo objetivo se recomienda agregar una tabla de horario macro, por ejemplo `CDS_AppointmentConfigScheduleLine`, con una estructura similar a `CDS_EmployeeScheduleLine`.

Campo sugerido	Uso
`CDS_AppointmentConfig_ID`	Configuración de agenda a la que pertenece el horario macro.
`WeekDay`	Día de la semana permitido.
`StartTime` / `EndTime`	Rango horario general de atención.
`ValidFrom` / `ValidTo`	Vigencia opcional del horario.

> **Importante:** si el diccionario cambia, se deben regenerar los modelos antes de ajustar el código Java.

Lógica Backend

Motor de disponibilidad, callouts, procesos y validaciones por evento.

Motor de Disponibilidad

`AppointmentAvailabilityEngine` concentra la lógica que decide si una cita o un slot es válido para un empleado, servicio y fecha determinados.

Responsabilidades actuales

- Cargar una configuración activa por organización con fallback a organización `0`.
- Leer duración, precio e intervalo del servicio.
- Aplicar overrides de precio y duración por empleado-servicio.
- Validar que el empleado pueda prestar el servicio.
- Validar días libres de la organización.
- Validar días libres del empleado.
- Validar cobertura del horario semanal del empleado.
- Detectar conflictos con citas existentes pendientes o aprobadas.
- Aplicar límites diarios por empleado cuando correspondan.
- Crear slots persistentes para la UI de iDempiere.

Regla de conflicto

Una cita se considera en conflicto cuando otra cita activa del mismo empleado se cruza con el rango nuevo y tiene estado pendiente o aprobado.

```
existing.StartDateTime < newEnd
AND existing.EndDateTime > newStart
AND existing.Status IN ('PE', 'AP')
AND existing.IsActive = 'Y'
```

Generación de slots

La generación actual es específica por empleado. El proceso no genera para todos los empleados ni autoasigna el recurso; parte de una selección explícita de empleado, servicio y fecha.

>

> **Nota:** esta decisión hace que el flujo de UI sea más claro: primero se elige quién atiende, luego qué servicio presta, después se generan horarios disponibles.

Reglas actuales vs reglas esperadas

Regla	Estado actual	Comportamiento esperado
Empleado presta servicio	Implementado con `CDS_EmployeeService`.	Se mantiene como regla obligatoria.
Día libre de organización	Implementado por organización. Aunque `CDS_OrgDayOff` tiene `CDS_AppointmentConfig_ID`, el motor actual no filtra por configuración.	Debe filtrar por configuración cuando el servicio resuelva una configuración específica.
Día libre de empleado	Implementado con `CDS_EmployeeDayOff`.	Se mantiene como bloqueo específico del empleado.
Horario del empleado	Implementado con `CDS_EmployeeScheduleLine`.	Debe validarse después del horario macro de la configuración.
Horario macro de agenda	Pendiente. No existe aún una tabla equivalente a horario de configuración.	Debe definir los días y rangos generales permitidos, por encima del horario del empleado.
Anticipación mínima	Pendiente. Existe `MinAdvanceHours`, pero no se valida todavía.	Debe bloquear citas creadas con menos horas de anticipación que las configuradas.
Anticipación máxima	Pendiente. Existe `MaxAdvanceDays`, pero no se valida todavía.	Debe bloquear citas demasiado futuras.
Conflictos con citas existentes	Implementado para estados `Pending` y `Approved`.	Se mantiene como validación obligatoria.
Aprobación automática o forzada	Pendiente. Existen campos de configuración y servicio, pero no se aplican en el evento.	Debe resolverse en `BeforeNew` usando configuración y servicio.

Roadmap del motor

- Resolver la configuración desde el servicio seleccionado, no solo desde la organización.
- Agregar validación de horario macro de configuración antes del horario del empleado.
- Aplicar `MinAdvanceHours` y `MaxAdvanceDays` al generar slots y al guardar citas.
- Filtrar días libres de organización por configuración cuando exista relación.
- Aplicar reglas de aprobación en el evento antes de validar el estado final de la cita.

Lógica Backend

Callouts y Flujo de UI

`CalloutAppointment` ayuda al usuario a capturar una cita desde iDempiere sin reemplazar la validación final del backend.

Comportamiento de callouts

Campo	Comportamiento
`CDS_AppointmentService_ID`	Copia precio y duración efectivos. Si existe empleado seleccionado, usa override de `CDS_EmployeeService`. Limpia slot y horas, pero no limpia empleado.
`AppointmentDate`	Limpia slot, hora inicial y hora final. No limpia empleado ni servicio.
`CDS_AppointmentSlot_ID`	Copia empleado, hora inicial, hora final, precio y duración desde el slot seleccionado.

Flujo recomendado en ventana

1. Seleccionar el empleado que atenderá la cita.
2. Seleccionar el servicio filtrado para ese empleado.
3. Seleccionar la fecha de la cita.
4. Ejecutar el proceso de generación de slots.
5. Seleccionar el slot generado.
6. Guardar la cita.

> **Importante:** si un usuario modifica manualmente horas, precio o duración, el evento de modelo vuelve a validar la consistencia antes de guardar.

Lógica Backend

Procesos y Eventos

El plugin mantiene pocos procesos activos y concentra las reglas críticas en eventos de modelo.

Procesos activos

Proceso	Uso
`GenerateAppointmentSlots`	Genera registros `CDS_AppointmentSlot` para el empleado, servicio y fecha seleccionados. Puede resolver valores desde la cita actual cuando se ejecuta desde la ventana.
`ReopenAppointment`	Reabre una cita procesada cambiándola a pendiente y limpiando `Processed` mediante un contexto controlado.

Proceso `GenerateAppointmentSlots`

Este proceso se usa para crear horarios disponibles persistentes que luego pueden consultarse desde la ventana iDempiere o desde la API leyendo la tabla `CDS_AppointmentSlot`.

Parámetro	Requerido	Uso
`AD_Org_ID`	Sí, directo o resuelto desde la cita	Organización para cargar configuración y validar disponibilidad.
`CDS_AppointmentEmployee_ID`	Sí	Empleado para el cual se generan slots.
`CDS_AppointmentService_ID`	Sí	Servicio que define duración, precio e intervalo del slot.
`AppointmentDate`	Sí	Fecha para la cual se generan horarios disponibles.

Parámetro	Requerido	Uso
`CDS_Appointment_ID`	Opcional	Si se ejecuta desde una cita, permite resolver organización, empleado, servicio y fecha desde el registro actual.

Antes de generar nuevos slots, el proceso desactiva slots previos activos del usuario actual. Si hay una cita, desactiva los slots asociados a esa cita. Si no hay cita, desactiva slots del mismo usuario, empleado, servicio y fecha.

> **Uso recomendado:** desde la ventana de cita se selecciona empleado, servicio y fecha; luego se ejecuta `GenerateAppointmentSlots`; finalmente se selecciona un registro de `CDS_AppointmentSlot`.

Proceso `ReopenAppointment`

Este proceso reabre una cita procesada para permitir correcciones administrativas controladas.

Parámetro	Requerido	Uso
`CDS_Appointment_ID`	Sí, directo o resuelto desde el registro actual	Cita procesada que se desea reabrir.

Si la cita ya está abierta, el proceso informa que no requiere reapertura. Si está procesada, activa un contexto temporal `#CDS_ReopenAppointment`, coloca `Processed=N` y cambia el estado a `Pending`.

Validación por evento actual

`AppointmentModelValidator` se ejecuta antes de crear o cambiar una cita. Para citas pendientes o aprobadas valida los datos mínimos y la disponibilidad real.

- Cliente requerido.
- Empleado requerido.
- Servicio requerido.
- Fecha de cita requerida.
- Hora inicial y final requeridas.
- Hora inicial menor que hora final.
- Fecha de cita igual al día de la hora inicial.

- Duración mayor que cero.
- Precio no negativo.
- Empleado habilitado para prestar el servicio.
- Slot consistente con empleado, servicio, horas, duración y precio.
- Sin bloqueo por día libre de organización o empleado.
- Horario cubierto por agenda del empleado.
- Sin conflicto con otra cita activa pendiente o aprobada.

Estados y procesamiento actual

- Si el estado viene vacío, se asigna `Draft`.
- Los estados finales marcan `Processed = true`.
- Los estados no finales limpian `Processed`.
- Una cita procesada no se puede modificar salvo mediante reapertura controlada.

Roadmap de validaciones en eventos

Las siguientes reglas deben agregarse al evento para que el comportamiento sea consistente desde ventanas iDempiere y desde REST/API.

Validación pendiente	Momento sugerido	Regla esperada
Resolver configuración desde servicio	`BeforeNew` y `BeforeChange`	El servicio debe indicar qué configuración de agenda aplica. La cita usa esa configuración para anticipación, horario macro y aprobación.
Servicio con una sola configuración	Al guardar servicio o relación configuración-servicio	Un servicio no debe estar asociado a más de una configuración activa.
Anticipación mínima	`BeforeNew` y generación de slots	Si `StartDateTime` está antes de ahora + `MinAdvanceHours`, la cita o slot debe bloquearse.
Anticipación máxima	`BeforeNew` y generación de slots	Si `StartDateTime` supera hoy + `MaxAdvanceDays`, la cita o slot debe bloquearse.

Validación pendiente	Momento sugerido	Regla esperada
Horario macro de configuración	`BeforeNew`, `BeforeChange` y generación de slots	El horario de agenda general debe contener el rango solicitado antes de evaluar el horario del empleado.
Forzar aprobación	`BeforeNew`	Si la configuración tiene `ForceApproval=Y`, la cita debe quedar pendiente aunque el servicio permita aprobación automática.
Autoaprobar si el servicio lo permite	`BeforeNew`	Si `AutoApproveIfServiceAllows=Y` y el servicio no requiere aprobación, la cita puede quedar `Approved`.

Roadmap de workflows y correos

Además de las validaciones, falta definir workflows y plantillas de correo para notificar al cliente cuando cambie el estado de una cita.

Flujo pendiente	Disparo sugerido	Resultado esperado
Notificar cita aprobada	Cambio de estado a `Approved`.	Enviar correo al cliente con fecha, hora, servicio y empleado.
Notificar cita cancelada	Cambio de estado a `Cancelled`.	Enviar correo al cliente con el motivo de cancelación cuando exista.
Notificar reprogramación	Cambio de fecha, hora o empleado.	Enviar correo con el nuevo horario confirmado.
Recordatorio de cita	Proceso programado antes de `StartDateTime`.	Enviar recordatorio al cliente según configuración.

También falta crear las plantillas de correo, definir destinatarios, configurar remitente, parametrizar cuándo enviar cada notificación y evitar reenvíos duplicados con campos de control si aplica.

Orden recomendado de validación

1. Resolver servicio, empleado y organización.

2. Resolver configuración de agenda desde el servicio.
3. Aplicar reglas de anticipación mínima y máxima.
4. Validar horario macro de configuración.
5. Validar días libres de organización vinculados a la configuración.
6. Validar relación empleado-servicio.
7. Validar horario y días libres del empleado.
8. Validar slot seleccionado.
9. Validar conflictos con citas existentes.
10. Resolver estado de aprobación: pendiente o aprobada.

Operación e Integración

Uso desde iDempiere, REST/API, instalación y 2Pack.

Operación e Integración

Uso desde REST/API

Los clientes externos pueden crear citas directamente en `CDS_Appointment` mediante la API REST estándar de iDempiere. No se requiere un proceso custom para crear la cita.

> **Idea clave:** REST/API y ventana iDempiere comparten la misma seguridad funcional porque el evento de modelo valida el registro antes de guardarlo.

Flujo recomendado

1. Consultar servicios activos en `CDS_AppointmentService`.
2. Consultar empleados activos y sus relaciones en `CDS_EmployeeService`.
3. Consultar o calcular disponibilidad usando la misma información de horarios y días libres.
4. Crear un registro `CDS_Appointment` con cliente, empleado, servicio, fecha, hora inicial, hora final, precio, duración y estado.
5. Dejar que `AppointmentModelValidator` valide consistencia, disponibilidad y conflictos.

Campos importantes

Campo	Uso
`C_BPartner_ID`	Cliente o tercero de la cita.
`CDS_AppointmentEmployee_ID`	Empleado que atiende.
`CDS_AppointmentService_ID`	Servicio solicitado.
`AppointmentDate`	Día operativo de la cita.
`StartDateTime` / `EndDateTime`	Rango exacto de atención.
`Price` / `DurationMinutes`	Valores efectivos de la cita, validados contra servicio/slot cuando aplique.

Campo	Uso
`R_Request_ID`	Referencia opcional/manual si una implementación quiere vincular una solicitud. El código del plugin no depende de este campo.

Operación e Integración

Instalación y 2Pack

La instalación requiere desplegar el bundle OSGi del plugin y aplicar el 2Pack correspondiente en iDempiere 12.

2Packs incluidos

Archivo	Uso
`2Pack_1.0.0.zip`	Paquete inicial del diccionario de aplicación.
`2Pack_1.0.1.zip`	Paquete actualizado con traducciones `es_CO` para campos, elementos, menús, ventanas, pestañas, procesos, parámetros, referencias y valores de lista.

Pasos de instalación

1. Compilar y desplegar el bundle `com.cdsoftware.appointment` en iDempiere 12.
2. Aplicar el 2Pack más reciente desde `com.cdsoftware.appointment/META-INF`.
3. Reiniciar o refrescar iDempiere para activar servicios OSGi, factories, anotaciones, callouts, procesos y eventos.
4. Configurar `CDS_AppointmentConfig` para la organización.
5. Crear servicios en `CDS_AppointmentService`.
6. Crear empleados en `CDS_AppointmentEmployee`.
7. Relacionar empleados con servicios en `CDS_EmployeeService`.
8. Configurar horarios semanales en `CDS_EmployeeScheduleLine`.
9. Configurar días no disponibles en `CDS_OrgDayOff` o `CDS_EmployeeDayOff`.
10. Probar la generación de slots y creación de citas desde la ventana.

> **Importante:** después de aplicar cambios de diccionario que modifiquen columnas o referencias, se deben regenerar modelos y revisar que el código siga usando las constantes generadas.